

Change Table:

Date of Change	Related Issue/Incident Number	Updated by	Revision
2/25/00	128, 101	F. Ernst	Update Application Overview, Reference Codes, Extraction File Layout, Exception File Layout, the Developer's Notes, and WDTIP Contact sections of the document to reflect changes made during the Code and Unit Test and System Test phases of the project
4/5/00	151	A. Leano	Add note to EDG that states explicitly the records in the extraction file must be a fixed length of 150 bytes. Attach the COBOL file layout for reference.
4/20/00	151	A. Leano	Add clarification to the EDG on the use of the Federal/State Only Flag in the Program Participation Transaction section of the Extraction File Layout.
4/20/00	151	A. Leano	Add note to EDG which explicitly states that WDTIP expects to receive one extract file per county per day.
4/20/00	151	A. Leano	Add note to EDG which explicitly states that batch version number starts at 0 (zero).
02/24/01		R. Frey	Add section to EDG for new transaction (LD01).
9/16/02	Inc #417 Inc #853 Issue #164	L. Holder	Add the following: Incident #417, Extenders; Incident #853, Vista Volunteer Exception Code; and Miscellaneous changes
12/20/02	803	K. Murdock	Add information regarding the new 48/53-60 Approaching Clocks Report.
6/15/03	Issue 231 Incident 876	K. Murdock	Add information regarding the new Repay Codes, 07/700 and 07/701.
01/12/04	Incident 930	G. Kajita/ L. Holder	Modified Extract File Layout for Diversion Transactions and Exception Processing Reason Code 208.
05/10/04	Incident 803	K. Murdock/ L. Holder/ G. Kajita	Updated the TRAC Reference Code, Transaction Navigation Code Information. Updated the table to reflect new Approaching Clocks Report.
05/10/04	Incident 830	K. Murdock/ L. Holder/ G. Kajita	Updated the TRAC Reference Code Note for Program Exception Reason Type Code. Provided information on program exception types of 02 and 05.
05/10/04	Incident 878	K. Murdock/ L. Holder/ G. Kajita	Corrected Extraction File Layout to correctly identify when an aid code is a mandatory field. Extraction File Overview Section made more consistent by renaming File Header to Batch Header and Trailer Record to Batch Trailer Record as identified in the Extraction File Layout. (Pages 40, 41, 81, 82)

05/10/04	Incident 921	K. Murdock/ L. Holder/ G. Kajita	Updated TRAC Reference Codes, Program Participation Type Code Effective 10/01/1999 for Aid Code 04/33, Aid Code 04/3H (F), Aid Code 04/3H(S).
05/10/04	Incident 932	K. Murdock/ L. Holder/ G. Kajita	Updated TRAC Reference Codes, Program Participation Type Code Effective 01/01/2004 for Aid Codes 32 and 3W.
05/10/04	Incident 940	K. Murdock/ L. Holder/ G. Kajita	Corrected Extraction File Layout for Federal/State Only Indicator and Program Discontinuance Reason Code from Mandatory to Mandatory/Optional. Updated Rule Information.

WELFARE DATA TRACKING IMPLEMENTATION PROJECT

EXTERNAL DEVELOPER'S GUIDE

Table of Contents

1. INTRODUCTION.....	6
BACKGROUND.....	6
SCOPE.....	6
ORGANIZATION OF THIS DOCUMENT.....	7
2. DEFINITIONS AND ACRONYMS	8
3. CONVERSION STRATEGY	13
CURRENT APPLICATION- PRE-SAWS.....	13
TRAC	15
CONVERSION OPTIONS	17
4. APPLICATION OVERVIEW	19
APPLICATION COMPONENTS.....	19
INFRASTRUCTURE OVERVIEW.....	22
DATABASE RESTRUCTURING.....	23
INITIAL/ONGOING LOADS	25
STATE DATA SOURCES.....	29
COMMON LOGIC PROGRAMS	33
SCREENS	36
REPORTS.....	38
5. DEVELOPER NOTES	39
5. DEVELOPER NOTES	40
EXTRACTION FILE OVERVIEW.....	40
TRANSMISSION PROTOCOLS.....	74
EXCEPTION FILE OVERVIEW.....	79
REPORTS OVERVIEW.....	87
6. DEVELOPMENT PROCESS.....	89
REVIEW OF SOURCE SYSTEM DATA.....	89
CREATION OF EXTRACTION/ LOADING PROGRAMS	90
TRIAL LOADS.....	91
INITIAL CONVERSION.....	91
ONGOING LOADS (AFTER INITIAL CONVERSION)	92
7. WDTIP TECHNICAL CONTACTS	93
8. ISSUES AND ASSUMPTIONS	94
ISSUES	94
ASSUMPTIONS.....	94
REQUIREMENTS VALIDATION (JUNE/JULY/ON-GOING).....	105
ATTACHMENT A – EXTRACTION FILE LAYOUT	109
INTRODUCTION.....	109

THE COBOL FILE LAYOUT IS PROVIDED FOR REFERENCE. ATTACHMENT B – EXCEPTION FILE LAYOUT	109
ATTACHMENT B – EXCEPTION FILE LAYOUT	110
INTRODUCTION.....	110
ATTACHMENT C – REPORT LAYOUT APPROACHING CLOCKS	111
INTRODUCTION.....	111
ATTACHMENT C.1 – REPORT LAYOUT 48/53-60 APPROACHING CLOCKS	112
INTRODUCTION.....	112
ATTACHMENT D – REPORT LAYOUT EXCEEDING CLOCKS	113
INTRODUCTION.....	113
ATTACHMENT E – REPORT LAYOUT MULTIPLE COUNTY AID	114
INTRODUCTION.....	114
ATTACHMENT F – REPORT LAYOUT MONTHLY PROJECTIONS	115
INTRODUCTION.....	115
ATTACHMENT G – TRAC DATA MODEL	116
INTRODUCTION.....	116
ATTACHMENT H – TRAC DATA DICTIONARY	117
INTRODUCTION.....	117
ATTACHMENT I – TRAC REFERENCE CODES	118
INTRODUCTION.....	118
 <i>Figure 1: Current Application.....</i>	 <i>14</i>
<i>Figure 2: TRAC Application.....</i>	<i>16</i>
<i>Figure 3: Application Components</i>	<i>21</i>
<i>Figure 4: Infrastructure Overview - TRAC Application.....</i>	<i>22</i>
<i>Figure 5: DB2 Database Change Process</i>	<i>23</i>
<i>Figure 6: DB2 Data Conversion</i>	<i>25</i>
<i>Figure 7: County Job Flow: Job ID B3DDL01</i>	<i>26</i>
<i>Figure 8: SCI Job Flow: Job ID B3DDL03</i>	<i>29</i>
<i>Figure 9: MEDS Job Flow: Job ID B3DDL02.....</i>	<i>31</i>
<i>Figure 10: Time Calculation Job Flow: Job ID B3DDL02</i>	<i>34</i>
<i>Figure 11: Agent Data Retrieval</i>	<i>35</i>
<i>Figure 12: Screen Program Flow.....</i>	<i>36</i>
<i>Figure 13: Monthly Reports Job Flow: Job ID</i>	<i>39</i>
<i>Figure 14: County to TRAC File Transfer</i>	<i>75</i>

1. Introduction

The purpose of this **External Developer's Guide** is to provide application developers reference materials for creating programs that will generate the batch extraction file through which counties will update the Tracking Recipients Across California (TRAC) application data. Additionally, this document will provide an overview of the transmission protocols through which all files will be sent, the methods used to load that data, and the process through which rejected data will be returned to the counties. Finally, this document includes a description of the reports the TRAC Application will generate.

Background

In response to the Personal Responsibility and Work Opportunity Reconciliation Act (PRWORA) of 1996, the State of California passed Assembly Bill (AB) 1542. AB 1542 institutes the Temporary Assistance to Needy Families (TANF) program in California and impose welfare time limits, as well as new programmatic and eligibility rules. In addition to welfare time limits, AB 1542 mandates work requirements through the CalWORKs program. As a result of the CalWORKs program, county welfare departments are required to have a mechanism to track eligibility time limits, and other related data on an individual level, across counties, and over time to comply with the tracking requirements of both State and federal mandates.

The purpose of the TRAC Application is, therefore, to provide a communication mechanism and central data repository that can be accessed by all technology-enabled counties and relevant agency systems in order to prevent welfare fraud and meet the requirements of Statewide Automated Welfare System (SAWS) legislation and the TANF and CalWORKs programs. It addresses the immediate need for federal and State Welfare Reform tracking requirements imposed by the Federal PRWORA, AB 1542 and relevant All County Letters issued by the California Department of Social Services (CDSS).

Scope

The scope of this **External Developer's Guide** document is limited to presenting an explanation of how the TRAC Application will work. Through the document, special attention is paid to the systems with which the TRAC Application will interface so the readers of this document can understand how to develop the programs that will interface with it. The intended readers of this document are business application developers external to the WDTIP project, that are responsible for developing programs and files that interface with the TRAC Application.

Organization of this Document

This document covers a wide variety of topics that readers will need to understand to successfully develop programs that will interface with the TRAC Application. This section describes the main components that make up the **External Developer's Guide**.

Each component is composed of many other subsections. The following is a high level description of the components that make up this document.

- ❑ **Definitions and Acronyms** : this section contains a list of definitions and acronyms to provide a clear definition of terms used throughout the body of the document
- ❑ **Conversion Strategy**: this section contrasts the design of the current application versus the TRAC Application. It explains how data flows into the TRAC Application and describes the different conversion strategies that will be used to load the initial data into the application
- ❑ **Application Overview**: this section describes how the programs within the application interact with one another and external systems. It describes the inputs and outputs of each program and the program's objective
- ❑ **Developer Notes**: this section contains the bulk of the information that external developers need to know to develop the programs that will interface with the TRAC Application. The following are the main components of this section:
 1. **Extraction File Overview**: includes descriptions and scenarios for each component of the Extraction File
 2. **Transmission Protocols**: explains the protocols that will be used between the county systems and the TRAC Application to send and receive data, such as the extraction file and exception reports
 3. **Exception File Overview**: includes description and scenarios for each component of the Exception File
 4. **Reports Overview**: explains the reports that will be generated from the TRAC Application
- ❑ **Development Process**: This section explains the process that will be followed to develop, test, and integrate the programs that will perform the system conversion and data loads into the TRAC Application
- ❑ **WDTIP Technical Contacts**: this section contains contact information for the WDTIP Team members that can offer technical support to the counties during their development efforts
- ❑ **Issues and Assumptions**: this section contains assumptions that have been recorded during the design phase of the TRAC Application
- ❑ **Attachments**: reference material that will be helpful to better understand the TRAC Application and the programs that will be necessary to interface with it. Examples of these attachments include file layouts, the TRAC Application data model, and WDTIP Reference Codes

2. Definitions and Acronyms

The following is a selected list of definitions and acronyms to provide a clear definition of terms used throughout the body of the document

- < Less Than
- < > Not Equal to
- = Equal to
- <= Less Than or Equal to
- > Greater Than
- >= Greater Than or Equal to

3270 & 3270 Emulation – A communication protocol between workstations and the HHSDC. An established communication link is referred to as a “session”. 3270 sessions connect dedicated terminals to the HHSDC. 3270 emulation is software driven session used to connect a workstation to the data center with a functionally equivalent interface.

Agent Subroutines – Programs responsible for retrieving data from database tables and manipulating/formatting data as needed before displaying it on the screens.

Architecture – The manner or structure in which hardware or software is constructed. It defines how a system or program is structured, how various components and parts interact, as well as what protocols and interfaces are used for communication and cooperation between modules and components which make up the system.

Backup – A process of creating a copy of a configuration item to prevent the loss of work.

Batch – Application programs that process on a predetermined schedule. In TRAC, all batch processing will be “background” processing.

BMS – Basic map support. A CICS facility for mapping the physical layout of a screen.

CICS – Customer Information Control System. An IBM system for online transaction processing.

CIN – Client Index Number.

COMMAREA – COMMunication AREA (COMMAREA) is used in online programs to pass data between programs.

Common Logic Subroutines – Programs responsible for processing functions used by multiple programs other than retrieving data from the database (i.e., error processing, screen help).

COPYBOOK – Contains either common logic (e.g., date validation) or common structure of working storage variables.

DASD – Direct Access Storage Devices (DASD) attached as peripheral devices to the

mainframe computer.

Database – A collection of interrelated data stored together in one or more computerized files.

DB2 – Database 2 (DB/2) is a relational database product from IBM. It operates in a mainframe environment with communication capability to many programming languages and to other database products operating in the same or other environments.

DB2 Subsystem – A single copy of DB2 running on the mainframe computer. Each subsystem has unique associated resources such as buffer pools and system catalogs.

DB2 Table – A logical entity internal to DB2 that defines the structure of a named collection of data attributes. Tables often correlate closely to the entities and attributes defined in the logical data model. Tables utilize the storage space defined in tablespaces and are contained entirely within them.

DB2 Tablespace – A logical entity internal to DB2 that represents defined physical storage within a database.

DCLGEN – DeCLarations GENerator(DCLGEN) is a program that creates embedded SQL DECLARE TABLE statements and corresponding COBOL structure declarations from table descriptions in the catalog.

Declare Cursor – SQL command used to retrieve more than one row of data from a database table.

DSNTIAR – Subroutine used to convert a SQL return code into a text message. DSNTIAR takes data from the SQLCA(SQL Communication Area), formats it into a message, and places the result in a message output area that is provided in the application program.

EXIT – Terminate a process.

Function – A defined objective or characteristic action of a system or component. For example, a system may have inventory control as its primary function.

Gigabyte – 1,073,741,824 bytes. A byte is a series of bits of a particular length, usually 8. Computer storage space is measured in bytes.

HHSDC – The Health and Human Services Data Center (HHSDC) is the location of the mainframe computers and the associated peripheral devices that support the proposed WDTIP application.

Internal Reader - The internal reader (INTRDR) function accepts JES3 output data as an input job stream. Any job can use the internal reader to pass a job stream to JES3. The internal reader allows jobs submitted from MVS/TSO/E SUBMIT command and jobs created as system output (SYSOUT=(class,INTRDR)) to be submitted to JES3 input service for processing. JES3 dynamically starts internal readers as needed.

Invoke – Initiate a program.

ISPF – Interactive System Productivity Facility. Full screen editor and dialogue manager.

JCL – Job Control Language. Commands used to control jobs and programs on IBM mainframe operating systems.

JES – The Job Entry System is a function within the operating system of mainframe computers that facilitates the execution of tasks, called JOBS, on the computer. These are multiple versions of JES: JES2 and JES3. JES3 is in use at the HHSDC.

JOIN – SQL command used when accessing data from multiple tables in a database.

LINK – Connects programs with the capability of returning control to the initiating program.

LINKAGE SECTION – Used to pass data between programs.

Mainframe – A computer or system of computers designed for the uninterrupted processing of data. Mainframe computers are designed to support the requirements of multiple users, or groups of users. The processing capacity of mainframe systems varies within broad parameters. Special environmental and security concerns often limit the physical placement of these systems.

May – Indicates an item or activity appropriate under some, but not all, conditions; for which there are a number of acceptable alternatives; or for which there is no professional consensus.

Map – The screen layout.

MOVE – Populate values into the data element/table column.

MVS – Short for Multiple Virtual Storage (MVS), the operating system for older IBM mainframes. MVS was first introduced in 1974 and continues to be used, though it has been largely superseded by IBM's newer operating system, OS/390.

Online – Application programs that run in the CICS environment. In TRAC, all online processing is “real time” processing.

OS/390 – An IBM mainframe operating system, featuring integrated MVS, UNIX, LAN, distributed computing and application enablement services through its base elements. These base services enable open, distributed processing and offer a foundation for object-ready application development. The OS/390 base includes a Communication Server that includes VTAM, the VTAM AnyNet feature, TCP/IP and TIOC. It provides SNA (3270), APPC, High Performance Routing, ATM support, sockets and RPC.

Passing – Moving values between application programs.

Peripheral Devices – Computing equipment connected by direct communication channels to a mainframe computer. Peripheral devices include disk drives, tape drives and printers.

Populate – Loading values in data fields.

READ – SQL command used to select column values from database tables or a program command used to read a file.

RECEIVE – In online screens, the Map is thrown from the screen to the CPU.

Referential Integrity – A set of relationships specifying a dependency between attributes in the data model. Referential Integrity rules can define parent – child relationships or require existence testing prior to an update operation. They can be defined within the database code or within the application logic that interfaces with the database.

RETRIEVE – Collect data from a database table or TSQ or COMMAREA.

SEND – In online screens, the Map is thrown from the CPU to the screen.

Shall – Indicates an item or activity is required.

Should – Indicates an item or activity is recommended.

SIS – SAWS Information System. The database currently being utilized by the Pre-SAWS application.

Spaces – Indicates when the alphanumeric field is empty.

SQL – Structured Query Language (SQL) is the programming language used to access data stored in DB2 databases.

SQL CODE 0 – Query was successful.

SQL CODE 100 – Row not found (i.e., no data found).

Stacked – Collect data and store in the TSQ.

TCP/IP – Transmission Control Protocol/Internet Protocol (TCP/IP) is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in the private networks called intranets and extranets.

TRAC – Tracking Recipients Across California.

TRAC Database – The database that will be created for the TRAC Application.

TRAC System – Refers to all components involved in TRAC.

TRAN-ID – 4-character transaction id, used in online programs to initiate screen access.

TSQ – Temporary Storage Queue, used to store bulk data.

VTAM – Virtual Telecommunications Access Method (VTAM) is the primary communication protocol between the HHSDC mainframe and terminals connected as either 3270 or 3270 emulation sessions.

Welfare Data Tracking Implementation Project (WDTIP) – The official name of the project that is the subject of this document.

WHERE – SQL command used for specifying a conditional query on the database.

Will – Indicates an item or activity is a goal, which may or may not be attainable. See Shall.

WRITE – Same as MOVE verb, but in programming semantics it is WRITE for TSQ.

XCTL – Connects programs and transfers control to the destination program.

XMIT – Transmit (shortened to XMIT for JES3 command syntax) is a mainframe function that allows files and information to be sent from one location to another using the TCP/IP network.

3. Conversion Strategy

The following section describes the conversion strategy for the TRAC Application. The current process for loading data into SIS (SAWS Information System), the Pre-SAWS database, is analyzed and contrasted with the proposed strategy for loading data into the TRAC database. Finally, the two proposed alternatives that were developed during the Joint Application Design (JAD) sessions are discussed.

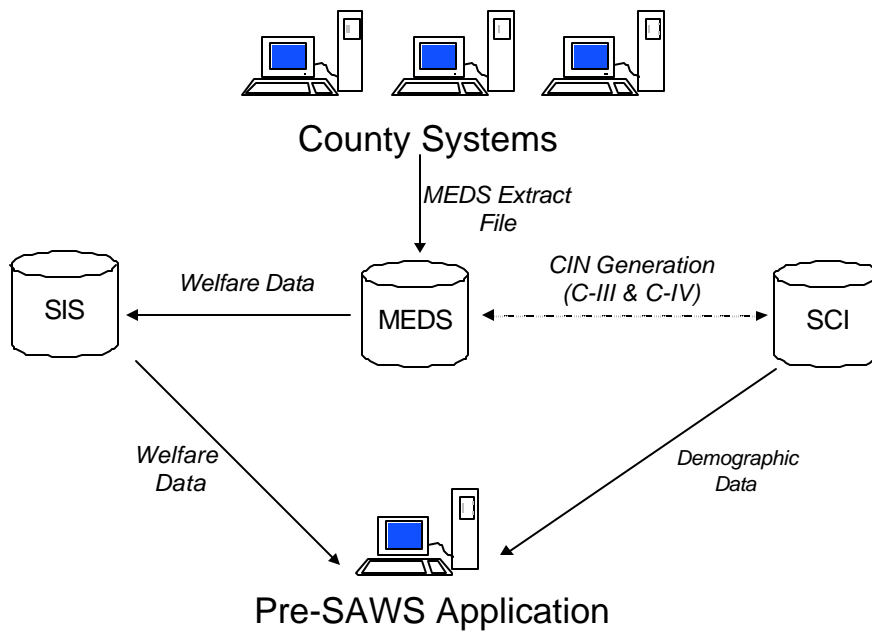
Current Application- Pre-SAWS

The current process begins when counties send an extraction file to MEDS containing all inserts and updates which occurred at the county systems over a specific period of time. If any of the records received by MEDS do not contain a Client Index Number (CIN), MEDS will interface with SCI to retrieve CINs for those records. After the counties extraction files are processed by MEDS, MEDS produces an extraction file that is sent to SIS containing a welfare related sub-set of the data sent by the counties. SIS processes the extraction file sent by MEDS and populates the database. After the data is loaded, the time clocks are calculated for each individual. Finally, before the data is displayed on the Pre-SAWS screens, the system interfaces with Statewide Client Index (SCI) to display demographic information such as the client's first and last name. The current system, however, is not meeting some of the requirements that are now demanded from the system:

- ❑ The data used to calculate the TANF-60 month, CalWORKs-60 month, and CalWORKs-18/24 month is not reliable because the data used by SIS to calculate the time clocks is not synchronized with the data in the county systems. Some data may be rejected through the MEDS process and never get sent to SIS
- ❑ Data elements for some functional areas that affect the calculation of the welfare time clocks are not currently being captured by SIS, such as Diversion and Exceptions information

The figure on the following page displays the flow of data into the current system.

Figure 1: Current Application



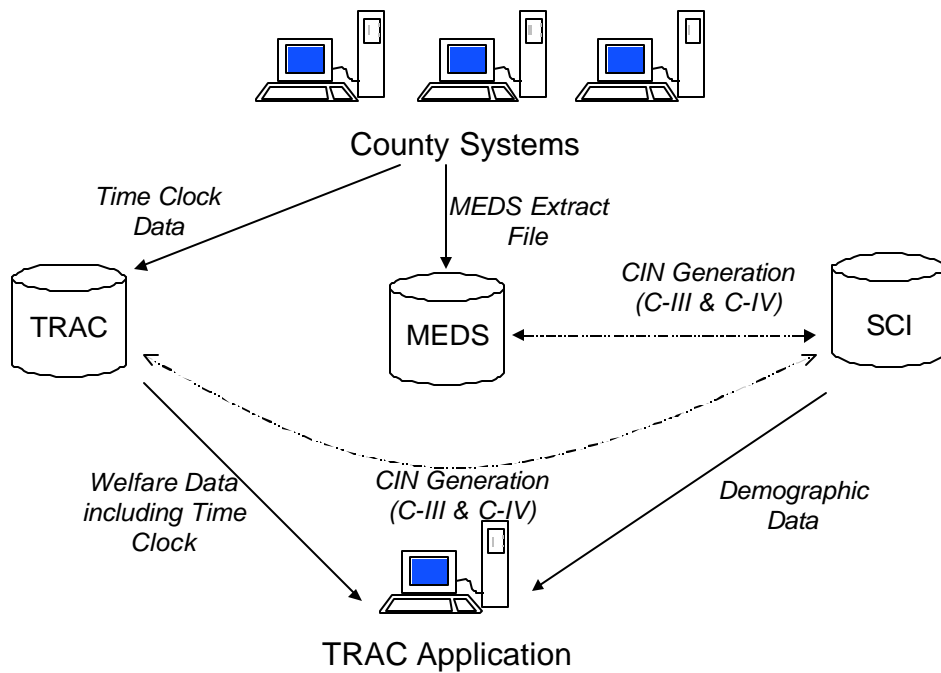
TRAC

- The objective of the TRAC Application is to capture mission critical data from source systems, namely county and consortia systems, and maintain the integrity of such data so that the data stored in its database is a replica of the data stored in the county and/or consortia systems. The TRAC Application will also collect additional data elements, which are not currently being captured by SIS such as Diversion information, and other data that is required to accurately calculate the CalWORKs 18/24-month, CalWORKs 60-month, and TANF 60-month time clocks.

The TRAC Application will still maintain its link with SCI through which it will obtain the demographic information about clients that is required for the screens display. Additionally, SCI will provide the TRAC Application directly with functionality to link an individual with a CIN (for counties that do not currently store CIN) as well as information on CINs that have been merged in SCI. This functionality is necessary to accurately reflect the information that SCI is providing to the counties and other State systems like MEDS.

The figure on the following page displays the flow of data into the TRAC Application.

Figure 2: TRAC Application



Conversion Options

For the initial conversion of SIS data into the TRAC Application, some of the existing SIS data will need to be modified to support the new calculation programs in the TRAC Application. One example of data that needs to be modified is the Participation Type Code that indicates if the individual is an adult or child. This value currently resides in the SIS database, but exists at an individual level. Since an individual can participate as a child in a case at one time in their life and as an adult at other times, the conversion programs need to determine the Participation Type Code for each program in which the individual participated. This determination is made by using a combination of other data, like Birth Date and Person Number.

Once the initial restructuring of the database has occurred, loading of county data can be initiated. The data that will be used by the TRAC Application will primarily be provided directly from the different county systems across the state. Currently, the data that exists in the SIS database, and that is used by the Pre-SAWS application, is obtained through an interface with MEDS. The initial conversion strategy of county data into the TRAC Application leverages the existing data in the TRAC Application and consists of three primary approaches. These three approaches have been labeled Option 1, Option 2 and Option 3. All three options try to address the complexity of converting data that is currently stored across multiple systems and formats into the TRAC database, and preserve the history of the existing SIS data that may not be available from the counties directly. The choice of option will be left to each county to decide.

Option 1

The first approach, Option 1, consists of populating data in the TRAC Application from the earliest start date that could be provided by the county, given that the earliest data is after 12/96. All data, which currently exists in Pre-SAWS (SIS database) from that point forward would be deleted from the SIS database and populated with the new data provided directly by the county. The data that exists prior to that point in time would remain in the future TRAC database (currently SIS) as it is currently found in SIS.

Option 2

The second approach, Option 2, consists of populating all data from the county systems back to 12/96 and deleting all county specific data on individuals. This option would be preferred for counties who can provide a complete history (back to 12/96) for individuals in their source systems because it would insure that at the time of conversion, the TRAC database would have a true replica of source system data.

Option 3

The third approach, Option 3, consists of populating all data in the TRAC Application from the county systems back to earliest data available by CIN and deleting all county specific data on individuals. All data, which currently exists in Pre-SAWS (SIS database) from that point forward would be deleted from the SIS database and populated with the new data provided directly by the county. This option would be preferred for counties who can

provide all data for individuals in their source systems as it would ensure that at the time of conversion, the TRAC database would have a true replica of source systems from the earliest date for which data is available by individual.

4. Application Overview

The objective of the *Application Overview* section is to provide a high level description of the Tracking Recipients Across California (TRAC) system, the components of the application, and how the components interrelate to provide applicant/recipient time clock related data. This overview will begin with a description of the TRAC Application addressing the following components:

- ❑ The TRAC database
- ❑ TRAC programs
- ❑ Data sources

The overview also provides a description of the following:

- ❑ How data will flow from external systems into the TRAC Application;
- ❑ How the database will be restructured;
- ❑ How initial/ongoing loads will populate the new database with data from the existing SIS database and MEDS;
- ❑ How common logic programs are used to provide functions used by multiple programs, and
- ❑ How screens and reports will be generated and used by Eligibility Workers.

Application Components

The TRAC Application is comprised of a DB2 relational database and four types of programs. The four types of programs are common logic, screens, reports, and initial/ongoing load programs. Additionally, TRAC receives data from four data sources that include the counties, MEDS, SCI, and the SIS database. The following is a description of the database, programs and data sources that make up the TRAC Application.

TRAC Database

- ❑ *TRAC Database*: The database is a DB2 relational database that will reside on the B3PD DB2 subsystem of the HHSDC mainframe (see the **System Architecture Model - Section 6** for more details).

TRAC Programs

- ❑ *Common Logic Programs*: Database access is provided to the online transactions through a series of agent subroutines, common logic subroutines, and copybooks containing processing logic (Refer to the *Definitions and Acronyms* section of this document for a list of definitions). Batch programs may also use these programs for database access. All programs are written using COBOL and SQL.
- ❑ *Screen (CICS) Programs*: The function of the screen programs is to display data on the TRAC application. Online transaction capability will be implemented using custom

programs written using COBOL and CICS. The TRAC Application provides online data inquiry as well as limited online data update capability via CICS transactions. Either 3270 or 3270 emulation connections between counties and the HHSDC will provide users with access to the application. Data that can be updated via batch processing will not be updateable online in the TRAC Application. Data that can be updated online will include Diversion, non-California program participation, Supportive Services Only and Child Support reimbursement information.

- ❑ *Report Programs:* Reporting will provide summary and detailed information from the system in formatted outputs. Reports will be generated through batch programs and transmitted electronically to the counties and CDSS for inspection and printing.
- ❑ *Initial/Ongoing Loads:* System data will be derived, in part, from county systems via batch processes. Counties will provide data in a common file format to support the initial conversion of their information into the system. This same common file format and transmission process will be used on a regular basis to update recipient data stored in the database.

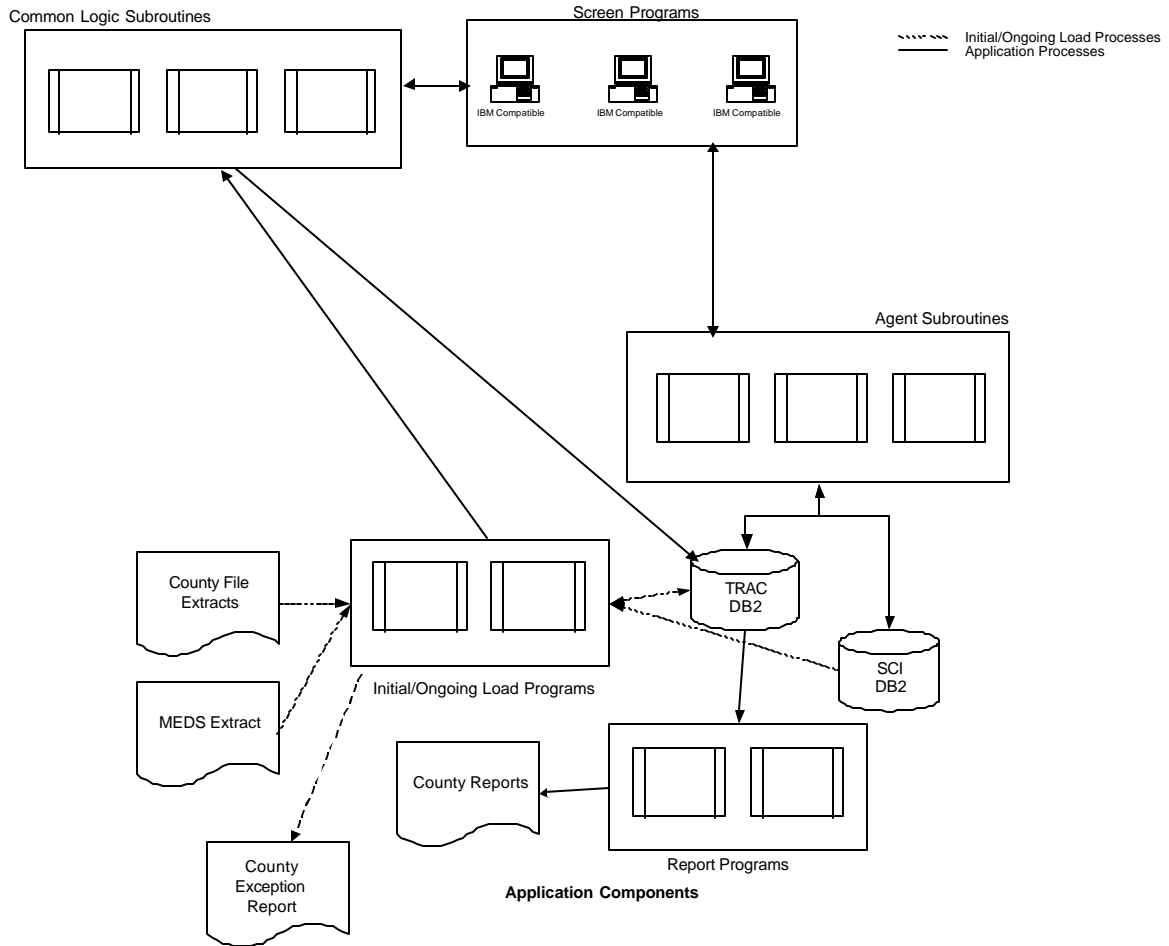
Data Sources

- ❑ *County Extraction File:* The application will interface with each of the county/consortia systems in the State of California. All counties will supply data to the batch load and conversion process; they will also have access to the online interface of the system.
- ❑ *MEDS Extract:* The application will interface with MEDS as the source of data for counties until they are converted and on an ongoing basis for non-time clock data elements.
- ❑ *SCI:* The application will interface with SCI for the generation and linking of CINs to records since some county systems do not retain this unique identifier.
- ❑ *SIS:* Data contained in the existing SIS will be converted to populate the TRAC Application database (TRAC).

The TRAC Application components are depicted in the following graphic:

Figure 3: Application Components

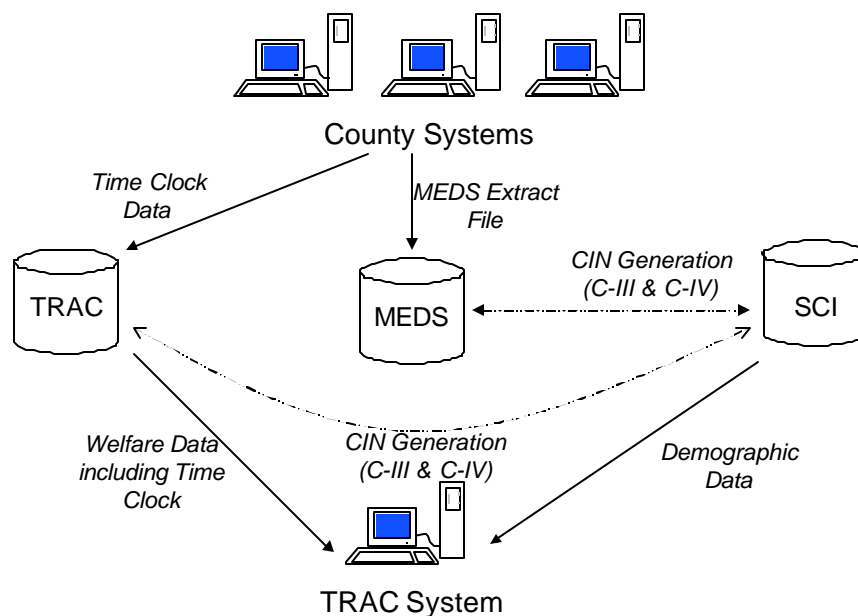
Application Overview



Infrastructure Overview

The TRAC Application is designed to accept all of the necessary time clock related data elements directly from the counties while relying on other statewide systems for some demographic data and other non-time clock related elements. These relationships are depicted in the following graphic.

Figure 4: Infrastructure Overview - TRAC Application



Time clock related data will be sent by each county in the standard extraction file format (described in the *External Developer's Guide* Appendix) to the TRAC Application. This file format will contain unique identifiers for each record that will be stored in the TRAC database and will link the information transferred to a specific individual. The CIN will be the key that TRAC will use to identify individuals. If a county system does not store CINs, the TRAC Application will send a batch file to the SCI database to identify the individual's CIN.

After the data has been loaded to the database, the TRAC Application will access programs that contain business rules associated with the calculation and display the data received. This common logic will also derive additional data elements that will subsequently be stored in the database.

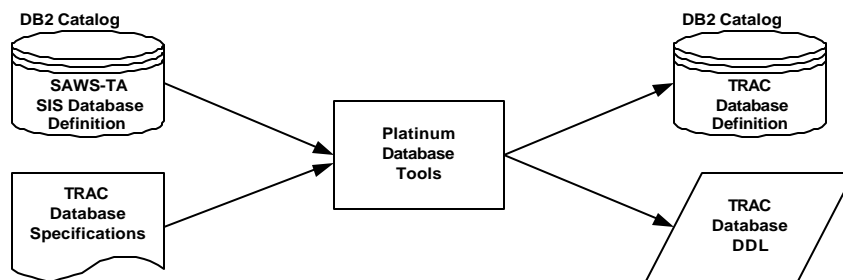
Database Restructuring

Enhancements to the current SIS database will be made to accommodate storage of time clock related data and assist in program data processing. The resulting database will be the TRAC database. The tasks required to enhance the database include:

- ❑ Adding or deleting entire tables
- ❑ Adding or deleting columns in existing tables
- ❑ Formatting changes to current fields

Implementing database modifications requires database management tools (Platinum) to reside on the mainframe. These tools are described in the **Tools** section of the **System Architecture Model**. Change definitions and the current database structure will be incorporated within the database management tools. Combining the change definitions and the existing database will result in a new database structure definition. This process is depicted in the graphic below:

Figure 5: DB2 Database Change Process



The DB2 Database Change Process is described in further detail in the following items:

- ❑ **SAWS-TA SIS Database Definition:** DB2 databases are defined through a series of interrelated definition tables collectively referred to as the DB2 Catalog. Within the Catalog are the details of the table and column attributes and the relationships between tables. The physical storage characteristics of the database are also defined within the Catalog. Each DB2 subsystem has a unique Catalog defining all aspects of the tables and databases that reside on that system. This item defines the existing database structure of the SIS database as it resides on the B3PD subsystem.
- ❑ **TRAC Database Specifications:** The written specifications for the TRAC database are contained within the **Database Design** section of this document. These specifications will be combined with the current DB2 definitions to create the new database.
- ❑ **Platinum Database Tools:** The mainframe tools in this suite will read the DB2 Catalog and allow the database administrator to modify the detailed characteristics of the database. The tools provide a formatted panel interface for the construction and

modification of DB2 objects such as databases, tablespaces, tables, columns, and indexes. The tools produce a change language file that is then used to modify the DB2 Catalog.

- ❑ *TRAC Database Definition:* This item is the DB2 Catalog entries resulting from the changes defined in the TRAC Database Specifications. This item defines the new structure of the TRAC database as it resides on the B3PD subsystem.
- ❑ *TRAC Database DDL:* Data Definition Language (DDL) is the coded instructions that produce changes in the contents of the DB2 Catalog. A file of DDL can be used as a backup for the structure of the database. The structure of the database can be recreated from the instructions contained in this file.

Initial/Ongoing Loads

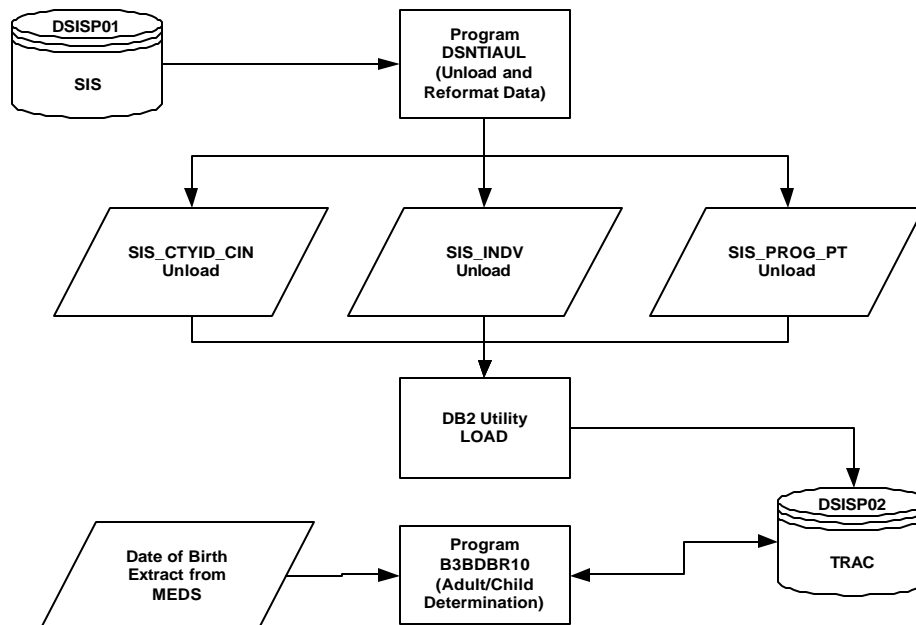
After the TRAC database is developed, the initial and ongoing data loads will serve to populate the new database. The steps required to load data include:

- ❑ Migrate data from SIS to TRAC
- ❑ Synchronize Data
- ❑ Load Data

Data migration from SIS to TRAC

Data migration begins with the program titled Unload and Reformat Data (DSNTIAUL) which will be used to migrate data from SIS to TRAC. This program is a supplied DB2 program that will be modified for the task of reformatting. For this reason, the program does not conform to WDTIP naming conventions. Data from the SIS database will be unloaded and reformatted into the format of TRAC. The data will then be loaded into the new DB2 structures using data load utility programs. A program titled Adult/Child Determination Program (B3BDBR10) will then update the date of birth and participation type code information in the database. This process is depicted in the graphic below:

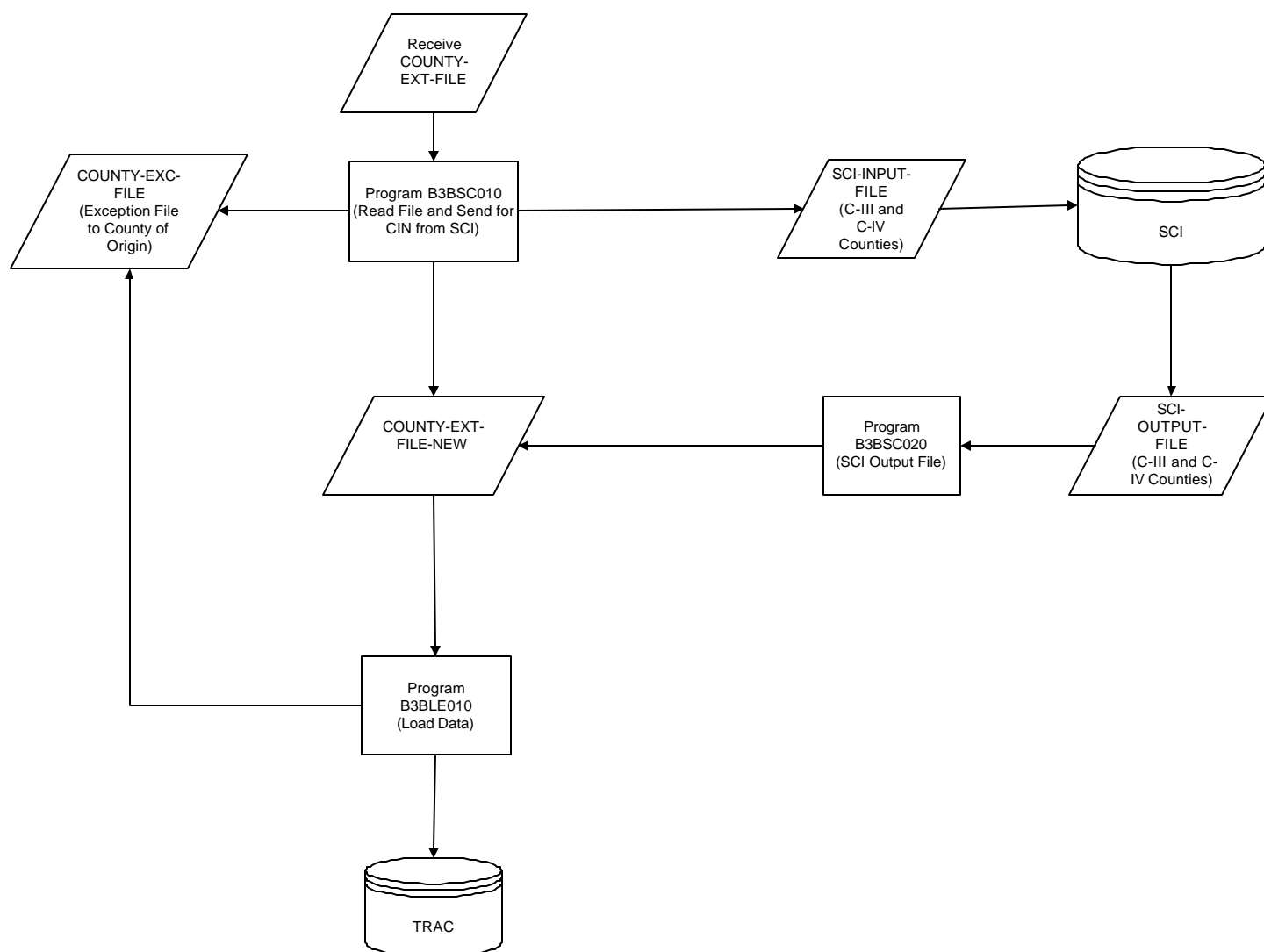
Figure 6: DB2 Data Conversion



Job B3DDC01

Once the existing data has been converted into the new format, that data must be synchronized with the data that comes directly from the counties. This will be initiated when each county transfers its respective extraction file (COUNTY-EXT-FILE) to TRAC. Only the specific data from the respective county will be updated. In other words, if the extraction file from Yuba County is received, only data that originated from Yuba County will be impacted. The process that the county data will follow is depicted in the following graphic:

Figure 7: County Job Flow: Job ID B3DDL01



After the COUNTY-EXT-FILE is received, a program will read the file and make a copy (COUNTY-EXT-FILE-NEW). This copy is made so that the integrity of the data is maintained in case any problems are incurred while the job is running. The name of this program is Create SCI Input File (B3BSC010). If the program encounters problems with the data in the Batch Header area of the COUNTY-EXT-FILE, a record will be added to the SIS_EXCEPTION_FILE table. Once the COUNTY-EXT-FILE-NEW has been created, the program determines if CINs are missing for each record. If a CIN is missing for a record, this program will search TRAC on the individual's county ID and attempt to see if that individual is known to the database. If the individual is found, the CIN is written into the copy of the extraction file. For any records that do not have CINs known to the database, a separate file will be created and sent to SCI. This file is titled SCI-INPUT-FILE. Although this file will be sent to an external system, this program is described in this section of the document because of its close dependency with the Initial/Ongoing Conversion Loads process.

SCI will use this file to link a CIN to a record or generate a new CIN through the SCI matching algorithm. This matching algorithm uses the individuals County ID, name, social security number, date of birth and gender to determine if the individual is known to SCI. A weighed scoring system is used to compare the incoming data with information in SCI and will then either associate an existing CIN with the record or generate a new CIN for the individual. SCI will attach the appropriate CIN to each record and send a file back to TRAC called SCI-OUTPUT-FILE. Once this file has been received, another program titled SCI Output File Program (B3BSC020) will attach these records to the COUNTY-EXT-FILE-NEW and send it on to be loaded to the database. In general, county data originating from a county that stores CINs in their respective eligibility system (e.g., ISAWS and LEADER) will not go through the matching process described in this paragraph.

Before the data is actually loaded to the TRAC database, the existing data provided from SIS will need to be synchronized with the new data. The factor that will determine how this process is completed will be identified by the conversion strategy option the county chooses to utilize. This option will be based on the accuracy and completeness of the data provided directly to the TRAC Application from each county. The conversion options include:

- ☐ Deleting all data that currently exists from the county (in the case that the county system has the capability to electronically provide that information directly)
- ☐ Deleting data after a point in time determined by the specific county (if the county can provide all data from that date forward for all its clients)
- ☐ Deleting data for each individual based on the earliest program participation effective date for that individual.

The programs that will read what option the county has chosen and synchronize the data are the Create SCI Input File (B3BSC010) if the county is using a single point in time, or the Extract Load Program (B3BLE010) if the county chooses to delete data based on a different date for each individual. This program will delete data based upon the information in the initial county extract received by TRAC. After this deletion process has taken place, a flag

will be set in the database to insure no further deletions will occur. This flag will be set after this program is run upon the initial load of data for each county.

Load Database

The program responsible for loading county data into the TRAC database is titled Extract File Load Program (B3BLE010). The first step in this program is to validate that all of the data sent by the counties (or generated by SCI) is complete and in valid format. If any of the data is not valid or is missing, the record will be written to the SIS_EXCEPTION_FILE table. Once B3BLE010 has run to completion, the County Exception File Program (B3BEX010) will create separate files for each county from all of the records on the SIS_EXCEPTION_FILE table. After that validation has occurred, the program will load data to tables in the TRAC database based on the specific transaction code provided by the counties. Each transaction will load data following a specific process. Please refer to the *External Developer's Guide* in the Appendix for examples of how the data is loaded.

As data is loaded, internal flags will be set in the Individual Table (SIS_INDV). These flags include indicators for:

- ☐ The existence of data in a table
- ☐ Calculation of the time clocks for that specific day
- ☐ Calculation of the time clocks at the end of the month

After all of the appropriate data has been loaded, a record of the specific transaction will be added to a table in the database (BATCH_TRX_LOG). The information stored in this table will be a unique identifier for the county file, when it was received, how many records were included in the file and how many records were processed.

State Data Sources

Apart from data received directly from the counties, the TRAC Application will interface with SCI and MEDS. SCI is referenced for the generation and linking of CINs to records for systems that do not store the CIN. MEDS will act as the source of data for counties until their data is converted and as the source of some non-time clock related data on an ongoing basis.

SCI

SCI will provide TRAC with information on CINs that have been merged in SCI. This merging process is necessary for individuals who may have had more than one CIN generated as a result of input errors or changes in their demographic data. When multiple CINs are identified, the counties notify SCI. SCI then links the CINs and assigns one as a “primary” and “freezes” the other(s). The TRAC Application utilizes the CIN Combination Update Program (B3BSC030) to update the TRAC database when a CIN merge occurs. B3BSC030 is depicted below in the graphic below.

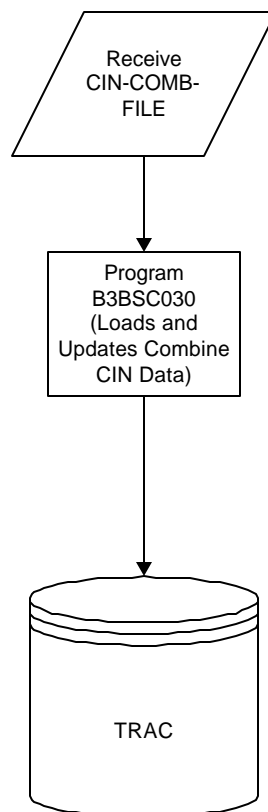


Figure 8: SCI Job Flow: Job ID B3DDL03

SCI sends a daily batch file (CIN-COMB-FILE) of merged CINs. B3BSC030 is initiated when CIN-COMB-FILE is received from SCI and checks if the merge is known to TRAC by referencing the SIS_CIN_COMBINATN table. If this record is known, the process will end. If there is a new merge, the program will add a record to the above table and then delete the record with the “frozen” or secondary CIN in the Individual table. The program will then delete information for the “frozen” CIN in the following other tables:

- ❑ Time Clocks (SIS_TIME_CLOCK, SIS_CALWRKS_18_DTL, SIS_CALWRKS_60_DTL, SIS_TANF_60_DTL)
- ❑ Welfare to Work (SIS_WTW)

Data is deleted for the frozen CIN(s) and the primary CIN will carry data for the individual's CINs.

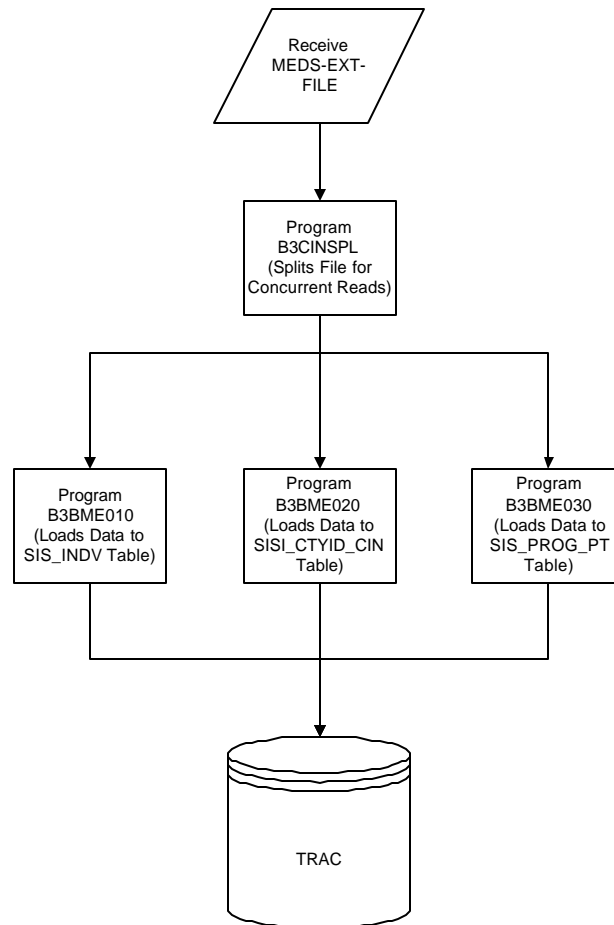
The program will then update all of the associated tables in the database that store the “frozen” CIN. By updating these tables, all data for that individual will be included in the time calculation process run daily by TRAC. As records are updated through the B3BSC030 program, a reference will be added to the SIS_CIN_COMB_AUDIT table. This table includes the new CIN and table name as well as the identifying information (CIN_SECONDARY, SECONDARY_CTY_CD, TABLE_NAME, and CREATED_TS) for the previous CIN.

Apart from this program, the TRAC Application will interface with SCI for the matching and generation of CINs. The programs that complete those tasks (B3BSC010 and B3BSC020) are described in the **Initial/Ongoing Loads** subsection of this document.

MEDS

Four programs will be used to load the data from MEDS into the TRAC database. The graphic below depicts the relationship between these programs.

Figure 9: MEDS Job Flow: Job ID B3DDL02



The job flow is initiated when a file is received from MEDS (MEDS-EXT-FILE). A program titled B3CINSPL will break this file into components that can be run concurrently. Since this program is currently being utilized by the Pre-SAWS application, its specifications are not included in this deliverable. Once the data has been split, three programs load data to separate tables in the database. The program titled Updated SIS_INDV Table Program (B3BME010) is a program that was modified from Pre-SAWS (hence, it does not conform to WDTIP naming conventions). This program populates data in the SIS_INDV table. This

program updates rows that are currently in the database by searching for a unique CIN. If the CIN is found, it updates the record; if it is not found, it will insert a new record. The program titled Update SIS_CTYID_CIN Table Program (B3BME020) functions similarly on the SIS_CTYID_CIN table.

The program titled Update SIS_PROG_PT Table Program (B3BME030) loads data to the program participation table for counties that have not yet begun sending loads directly to the TRAC database. In addition to directly loading data, this program will populate some information that is not provided by MEDS, namely the Participant Type Code. This program will use the date of birth as well as person number and county to generate a value for the program participation record. This logic is based on the county of origin, person number (in most cases), and the individual's age. Once all of the counties are converted, this program will be discontinued.

It is important to note that both the complete MEDS Job Flow and the County Job Flow must be completed prior to the initiation of the SCI Job Flow (discussed in the ***Initial/Ongoing Loads*** subsection previously).

Common Logic Programs

The TRAC Application incorporates common logic programs that are called by other programs within the system. The types of common logic programs are common logic subroutines, agent subroutines, and copybooks containing processing logic. Common logic subroutines provide functions such as time clock calculation and diversion calculation to other programs in the system. While agent subroutines are accessed by screen programs for data retrieval from the TRAC database and SCI. Copybooks that contain processing logic provide common functions such as error processing, help screens, and PF Key functionality.

Common Logic Subroutine - Time Calculation

Once data has been loaded into the database and combined CINs have been updated, the TRAC Application will initiate the Time Calculation Process. This program is run daily and will calculate the time clocks for each individual whose record was updated since the last time the program was run. The program will then reference all of the appropriate tables in the database and generate values for the following tables:

- ❑ Time Clocks Summary (SIS_TIME_CLOCK)
- ❑ Welfare to Work 18/24-Month Clock (SIS_CALWRKS_18_DTL)
- ❑ CalWORKs 60-Month Clock (SIS_CALWRKS_60_DTL)
- ❑ TANF 60-Month Clock (SIS_TANF_60_DTL)

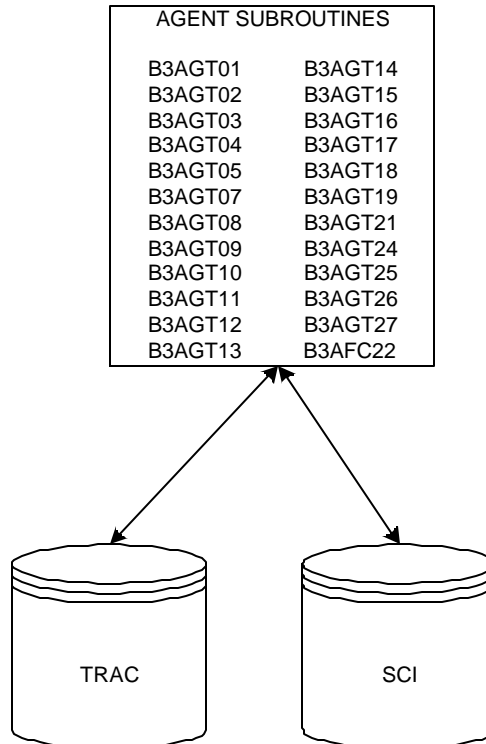
This process also will be initiated for all individuals who have an open program participation record that impacts the time clocks. This program will be run on the first day of the month based on the monthly recalculation flag that resides in the SIS_INDV table. This process is depicted in the graphic below.

Figure 10: Time Calculation Job Flow: Job ID B3DDL02

Agent Subroutines

Agent subroutines retrieve data for display on screens. Agents process requests and retrieve data from appropriate tables in the TRAC database. Multiple screens may access the same tables for data and use the same agent for their data retrieval. This process is depicted in the following graphic.

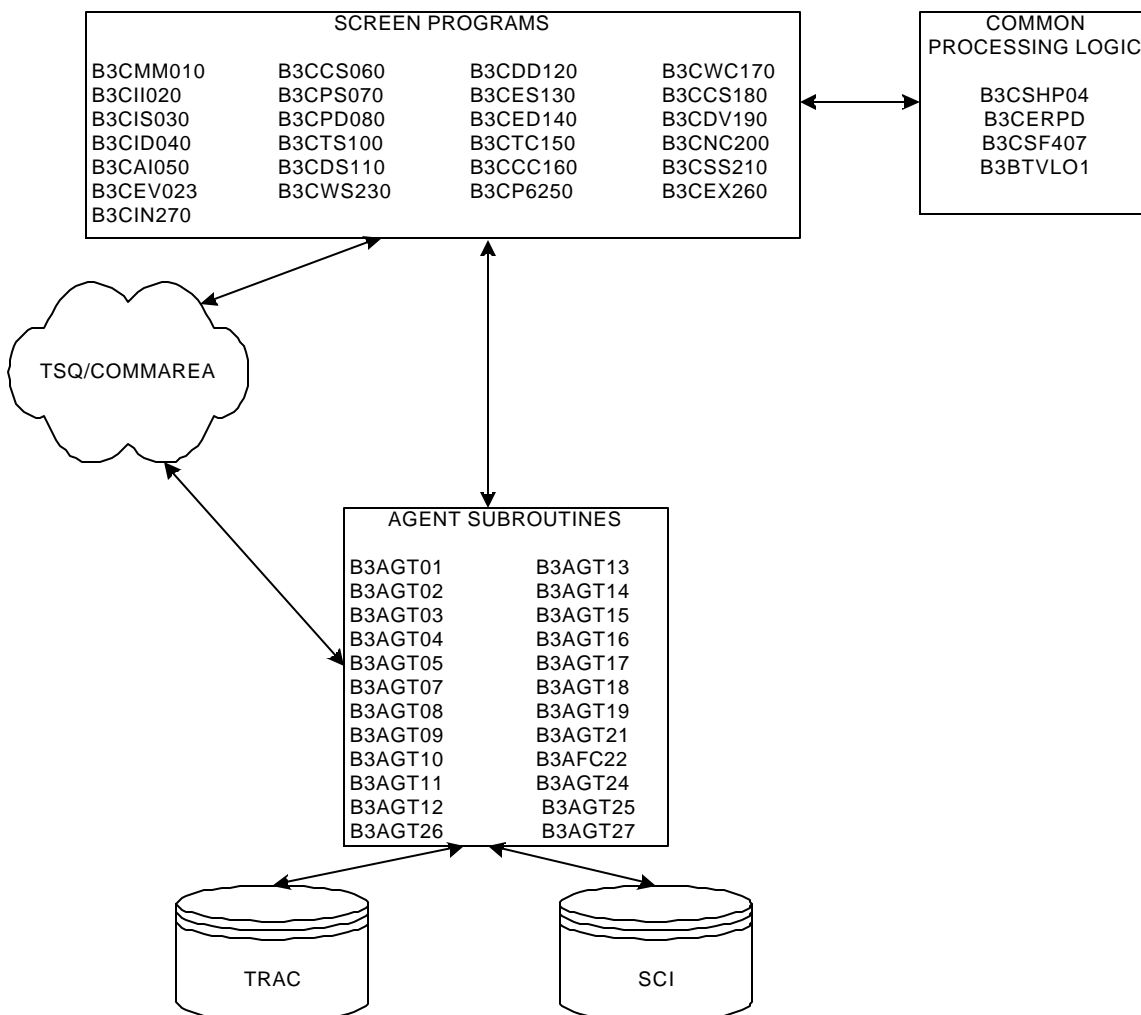
Figure 11: Agent Data Retrieval



Screens

Once the database has been populated, the TRAC Application will provide access to data through a series of screens. Data is displayed on screens using a screen/ agent method. In this method, screens are coded to display information, and agents perform data retrieval. This method is used to reduce redundant data retrieval logic for screens that display data from the same database tables. The screen/ agent method is depicted in the graphic below.

Figure 12: Screen Program Flow



Generally, the following process occurs when a screen is accessed:

1. All screen programs utilize an agent subroutine for data retrieval.
2. Agent subroutines then retrieve data from the TRAC database and SCI and populates a temporary storage area (COMMAREA or TSQ) depending on the requirement for the screen.
3. The screen then retrieves the data from the COMMAREA or TSQ for display.

Reports

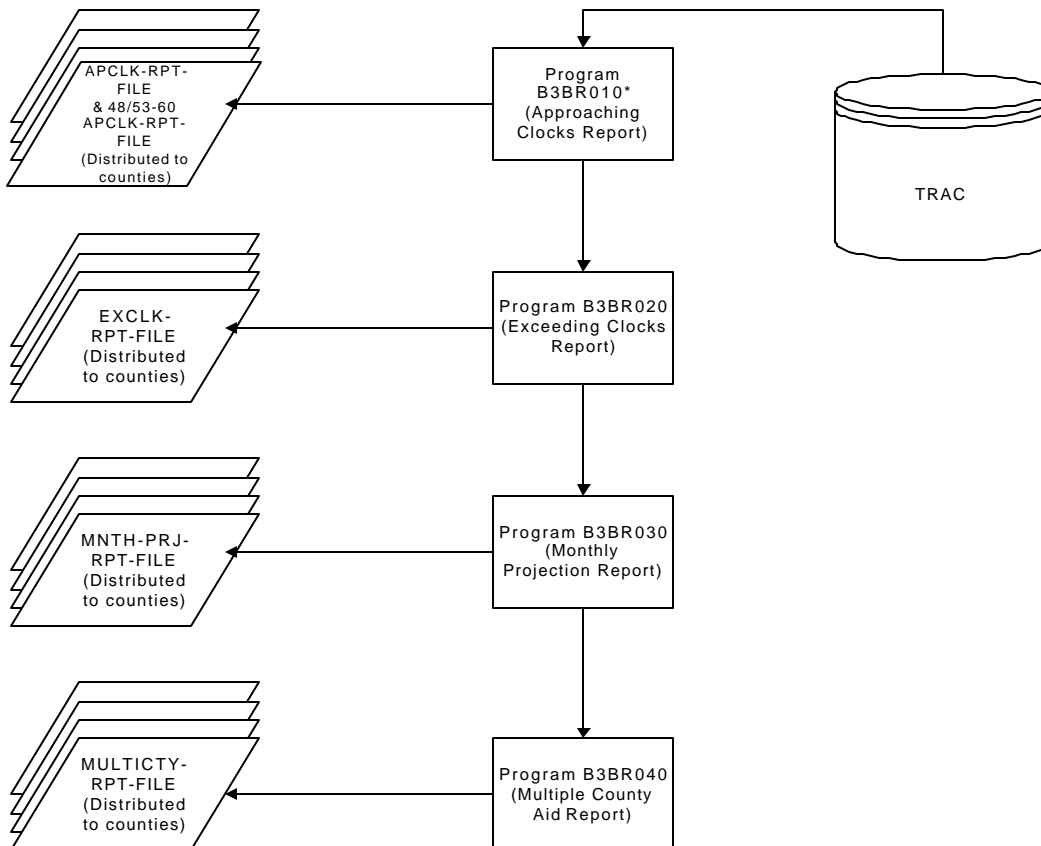
In addition to screens that provide access to data, the TRAC Application provides a series of reports to the counties and CDSS in electronic format. The users of the TRAC Application will be provided with statewide, aggregate data that reflects the information contained in the counties' individual eligibility systems.

Once the monthly calculation of time clocks is complete, the following report files will be generated.

- ❑ **Approaching Clocks Report** (APPCLK RPT FILE) - This is an existing report file which generates a summary (provided to CDSS) and detailed list (provided to the counties) of active individuals, by CIN or County ID, who are in month 54 or month 58 of the 60 month time limit for the State (CalWORKs) and/or federal (TANF) Programs; and/or in month 15/21 of the 18/24 month time limit for Welfare to Work. Counties may continue to receive this report file until they begin receipt of the new 48/53-60 Approaching Clocks Report (listed directly below).
- ❑ **48/53-60 Approaching Clocks Report** (48/53-60 APPCLK RTP FILE) – This new report file will generate a summary (provided to CDSS) and detailed list (provided to the counties) of active individuals, by CIN or County ID, who are in one of the following months of the 60 month time limit for the State (CalWORKs) and/or federal (TANF) Programs: 48, 53, 54, 55, 56, 57, 58, 59, or 60; and/or in month 15/21 of the 18/24 month time limit for Welfare to Work.
- ❑ **Exceeding Clocks Report** (EXCLK-RPT-FILE) - A summary (provided to CDSS) and detailed list (provided to the counties) of active individuals who have exceeded the 60-month time limit on the State (CalWORKs) and/or federal (TANF) limits; and/or active individuals who have exceeded the 18/24-month Welfare to Work time limit.
- ❑ **Monthly Projections** (MNTH-PRJ-RPT-FILE) - A report that provides a monthly projection of the **number** of active individuals who are in one of the following months of their State (CalWORKs) and/or federal (TANF) 60-month time limits: month 54, 55, 56, 57, 58, 59; and/or the number of active individuals who are in one of the following months of their Welfare to Work 18/24 month time limit: month 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23.
- ❑ **Multiple County Aid** (MULTICTY-RPT-FILE) – A report that contains the number of recipients who have received aid in more than one county over a lifetime of aid.

The TRAC Application will generate five report files each month containing the data for the above reports. The data for the reports will be consolidated into five files which group similar data into each file to minimize the number of reports distributed each month. Report files for each of the 58 counties and summaries for CDSS will be generated as well as stored for future access. The reports will be stored for three cycles before being overwritten with new reports. This process is depicted in the following graphic.

Figure 13: Monthly Reports Job Flow: Job ID B3PMR01, B3PMR02, B3PMR03, B3PMR04



*This Program produces two approaching clock reports.

5. Developer Notes

The following section lays out the components that external developers need to understand in order to develop programs that will interface with TRAC. The following topics are explained in detail in this section:

- ❑ Extraction File Layout
- ❑ Transmission Protocols
- ❑ Exception File Processing
- ❑ Reports Overview

Extraction File Overview

The extraction file layout dictates the structure for the batch file that counties/consortia will send to the TRAC database in order to insert and/or update data in TRAC. The file layout is composed of the following four components:

- ❑ Batch Header
- ❑ Batch Trailer Record
- ❑ Record Header
- ❑ Transactions

As a general note, the extraction file should only contain transactions for which an insert or an update occurred in the county system. Counties should not send transactions to the TRAC Application unless there has been a change to the data being sent in the county system.

The remainder of this section focuses on describing each of the four components of the extraction file, focusing on areas such as exception handling and load scenarios. For definitions of the data elements that make up each component of the extraction file, categorization of these data elements as mandatory or optional, and the ranges of acceptable values for each field, refer to ***Attachment A – Extraction File Layout***.

Batch Header

The Batch Header marks the beginning of the extraction file and contains general identifying information about the batch file being received, such as the county and the day for which that file contains data. It is imperative that all extraction files sent to the TRAC Application begin with a Batch Header. For the Batch Header, the failure to meet any of the following two business rules would cause the rejection of the whole extraction file:

1. The Batch Header section is missing from the county extraction file.
2. Any of the mandatory fields included in the Batch Header are missing or contain a value that is not accepted by TRAC.

Batch Trailer Record

The Batch Trailer Record marks the end of the extraction file and, like the Batch Header, contains general information about the batch file being received. They are different in that the Batch Trailer Record includes an extra field, which displays the count of the total number of records that are contained in the file. All extraction files sent to the TRAC Application will need to end with a Batch Trailer Record. For the Batch Trailer Record, the failure to meet any of the following two business rules would cause the rejection of the whole extraction file:

1. The Batch Trailer Record section is missing from the county extraction file.
2. Any of the mandatory fields included in the Batch Trailer Record are missing or contain a value that is not accepted by TRAC.

Record Header

The Record Header contains information about the individual transactions that update the TRAC database. It contains individual identifying data such as the CIN, Case Serial Number, Case FBU MEDS Code, and Person Number. Including the Record Header in each transaction will allow the TRAC Application to process transactions independently since the CIN and the County Code (Case Serial Number, Case FBU MEDS Code, County ID and Person Number) will be stored in the TRAC database tables for linking records with transactions containing inserts/updates. Additionally, the Record Header, as well as the Batch Header and the Batch Trailer Record, contains identifying information about the extraction file received such as the County Code, Batch Number, and Transaction Version.

If the Record Header fails to meet the following business rules, the whole Record Header and the transaction contained in it will be rejected.

1. All transactions contained in the extraction files will need to start with a Record Header.
2. If the county's system utilizes the CIN as an individual's unique identifier, then CIN and Case Serial Number, at a minimum, will be considered as mandatory fields for that county. Additionally, the TRAC Application will be expecting as mandatory fields for that county any combination of Case FBU MEDS Code and Person Number that is used by a specific county, to uniquely identify an individual within the county. The rest of the fields that are not necessary to uniquely identify an individual within a county will be optional. The exclusion of optional fields in the Record Header will not result in the record being rejected.
3. If the county's system does not utilize the CIN as an individual's unique identifier then, all of the following values will be considered as mandatory, with the exception of SSN, for that specific county's Record Header. These are necessary to interface with SCI's matching algorithm in the CIN retrieval process.

- ☐ Case Serial Number
- ☐ Case FBU MEDS Code
- ☐ Person Number
- ☐ Last Name

- ☐ First Name
 - ☐ Middle Name
 - ☐ SSN
 - ☐ Gender
4. If the county uses Aid Code to uniquely identify the record (for example, Ventura County), Aid Code should be included as a mandatory field; otherwise, counties should not include Aid Code in the Record Header.
 5. If any of the required mandatory fields for a specific county contains data that does not meet the data validation rules the record will be rejected.

Important Note: The field Caseworker ID is an optional field that has been included in the extraction file layout (refer to *Attachment A- Extraction File Layout*) to tie exception records with caseworkers in a county. The field will not be stored in the TRAC database, and will only be used to append to the rejected rows contained in the exception file to assist counties with the data cleansing process. If the field is not provided, the record will not be returned. Because the field is optional and is not stored in the database, there will be no validation checking for acceptable values for this field; therefore, no records will be rejected based on the value provided for the Caseworker ID.

Transactions

Transactions represent the main mechanism used to load data from the county extraction file into TRAC. The TRAC Application will also load data through update screens; however, updates will be limited to those functional areas for which counties are not storing the data in an electronic format. Those functional areas where updates will be available through screens include:

- ❑ Diversion
- ❑ Non-California Participation
- ❑ Child Support Reimbursement
- ❑ Supportive Services

The extraction file should only contain transactions for which an insert or an update occurred in the county system. Counties should not send transactions to the TRAC Application unless there has been a change to the data being sent in the county system.

The rest of this section provides scenarios for the different transactions that will load data into the TRAC database. For each transaction, the following subsections will be included:

- ❑ Description: explaining the purpose of the transaction
- ❑ Business logic: explaining the high level logic that is used to load each specific transaction into the TRAC database
- ❑ Scenarios: showing examples of how different transactions load into the TRAC database

In each example scenario, for the sake of simplicity, all of the mandatory data elements that are part of a transaction have not been included. Please refer to **Attachment A - Extraction File Layout** to identify the mandatory elements that **must** be included in each transaction.

Important Note: Although the value 'null' is included in the scenario examples, DB2 does not efficiently store 'null' values. When the TRAC Application is implemented, a value representing 'null' will be used (i.e.: the value 99991231 could be used to represent end dates).

Transaction Date of Birth Change

Description

This transaction is designed to update data in the individual table (SIS_INDV). The data contained within the record header of the transaction will be used to identify the record that is being updated. The DOB field will be updated using the date of birth sent in the Birth Date field of the extract file.

The business logic outlined below is applicable to all counties.

Business Logic

1. All data that is added to this table by a county will only affect other data for that county. The county where the data originated will always be the “owner” of the data
2. The CIN will be used as a **Key** to identify records in the database.
3. Through the use of the **Key**, this program will reference the CIN (provided by the county, or provided by SCI if the county is a non-CIN county) to identify the individual and update the data associated with individual in the database.
4. If the **Key** locates a row where the CIN contained in the transactions is equal to a row in the database, the Date of Birth contained in the transaction will replace the Birth Date in the database
5. If the **Key** does not locate a row where the CIN contained in the transactions is equal to a row in the database, a new record will be inserted into the Individual Table.

Scenarios

The following scenario has been identified to demonstrate, at a high level, the logic used to update an individual's date of birth in the TRAC database. This section will use the Database Image Table to provide before and after images that show how each transaction impacted the TRAC database. The following scenario assumes that row #1, in the Database Image Table, already existed in the TRAC Application prior to receiving the transactions that are described in the following scenarios.

Scenario #1

This transaction will demonstrate how to update a record in the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search for the Key following the logic described in Business Logic 4
- ❑ A matching row is not found (Program Type Code was different in the Key). The program uses Business Logic 5 and inserts a new row into the database with a Program End Date equal to null

Transactions Received Table

Data Elements	Transaction 1
CIN	123456789
Case Serial Number	1234567
FBU	1
Person Number	01
Date of Birth	19760607

Database Before Image for Transaction 1

Row #	CIN	DOB
1	123456789	19770607

Database After Image for Transaction 1

Row #	CIN	DOB
1	123456789	19760607

Transaction TRAC-Program Participation

Description

This transaction is designed to load data to the program participation table (SIS_PROG_PT) through the use of effective dates and status codes. These effective dates and status codes will build periods of eligibility for an individual with start and end dates that will be used by the program logic to do transaction operations such as inserting and updating records for an individual. The periods created by the transactions will then be used in the calculation of an individual's time clocks.

The business logic outlined below is applicable to all counties except Riverside and San Bernardino.

Business Logic

1. All data that is added to this table by a county will only affect other data for that county. The county where the data originated will always be the "owner" of the data
2. The County Code, Aid Code, Case Serial Number, FBU MEDS Code, Person Number, CIN, Program Type Code and Federal/State Only Indicator will be used as a **Key** to identify records in the database
3. Through the use of the **Key** this program will reference the Program Effective Date and Program Status Code provided by the county to identify specific periods of eligibility and update the data associated with those periods in the database
4. If the **Key**, Program Effective Date and Program Status Code locates a row where the Program Effective Date falls before the Program End Date, and the Program Status Code is equal to 'Aid', the Program Effective Date contained in the transaction will replace the Start Date in the database
5. If the **Key**, Program Effective Date and Program Status Code locates a row where the Program Effective Date falls after the Program Start Date, and the Program Status Code is equal to 'Disc', the Program Effective Date contained in the transaction will replace the End Date in the database
6. If the **Key**, Program Effective Date and Program Status Code do not locate a row where the Program Effective Date falls before the Program End Date, and the Program Status Code is equal to 'Aid', a new period will be inserted with a program end date defaulted to a value representing null
7. The Program Status Code can end one period of eligibility (when Program Status Code equals 'Disc' and begin another (when Program Status Code equals 'Aid')
8. When the Program Effective Date is added as a Program Start Date, all the data (like Aid Code, Participant Type Code, Minor Parent Flag, etc.) associated with the Program

- Effective Date (from the county) will be added to the database in that same record. The one exception to this rule is Program Discontinuance Reason Code
9. When the Program Effective Date is added as a Program End Date, the only associated field that is added to the row in the database is the Program Discontinuance Reason Code
 10. In some circumstances, the data that the county has sent may not be added because of the current condition of the data (i.e., a transaction with a Program Status Code of 'Disc' has been sent where an associated 'Aid' record does not exist). These instances will be included in the Exception File and sent back to the county
 11. Program Discontinuance Reason Code is an optional value, unless the transaction has a Program Status Code of 'Disc'
 12. Anytime the Program Status Code equals 'Aid', there should be no Program Discontinuance Reason Code populated in that record
 13. The Program Effective Month is an optional value that is used to synchronize cycle periods of eligibility with monthly periods of eligibility. This field is mandatory for Fresno County only. All other counties may leave this field blank
 14. To correct a Program Start Date for a period that has a Program Start Date, a transaction must be sent with a Program Status Code of 'Aid', the correct Program Effective Date, and the **Key**. Since this Program Effective Date will populate the database as a Start Date, all of the associated field values (like Aid Code, Participant Type Code, Minor Parent Flag, etc.) must be included in that transaction
 15. In order to send a complete program participation period (with Program Start and Program End Dates) the county will need to send two separate transactions. The first transaction will set the Program Start Date (with all associated fields), and the second transaction will set the Program End Date (with the Program Discontinuance Reason Code)
 16. Counties should not send any minor consent records to the TRAC Application
 17. When sending Aid Codes 3E, 3U, and 3H (3H – effective 10/1/99) counties should send include a value in the optional field Federal/State Only Flag indicating the clock which the Aid Code will tick. Please refer to **Attachment A- Extraction File Layout** for a list of valid values.

Scenarios

The following three scenarios have been identified to demonstrate, at a high level, the logic used to load program participation related data into the TRAC database. This section will use the Database Image Table to provide before and after images that show how each transaction impacted the TRAC database. The following scenarios assume that row #1, in the Database Image Table, already existed in the TRAC Application prior to receiving the transactions that are described in the following scenarios.

Note: The way the described scenarios impact the database can be followed in each scenario by comparing the before and after database images. Additionally, some data elements, like Participation Type Code and Minor Parent Flag, have not been included in the examples for the sake of simplicity.

Scenario #1

This transaction will demonstrate how to insert a new record into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search for the Key following the logic described in Business Logic 3
- ❑ A matching row is not found (Program Type Code was different in the Key). The program uses Business Logic 7 and inserts a new row into the database with a Program End Date equal to null

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 1</i>
CIN	123456789
Case Serial Number	1234567
FBU	1
Person Number	01
County Code	05
Program Effective Date	19980101
Program Type Code	04
Aid Code	3H
Federal/State Only Indicator	N
Program Status Code	Aid

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>County Code</i>	<i>Aid Code</i>	<i>Program Type Code</i>	<i>Program Start Date</i>	<i>Program End Date</i>
1	123456789	1234567	1	01	05	90	19	19980101	Null

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>County Code</i>	<i>Aid Code</i>	<i>Program Type Code</i>	<i>Program Start Date</i>	<i>Program End Date</i>
1	123456789	1234567	1	01	05	90	19	19980101	Null
2	123456789	1234567	1	01	05	3H	04	19980101	Null

Note: For 04/3H, a DISC transaction effective 19990930 should be sent to end date the period where the Federal/State Only indicator is not required. An AID transaction with the appropriate Federal/State Only indicator effective 19991001 should be sent to designate the appropriate funding source.

Scenario #2

This transaction will demonstrate how to change the Aid Code for an individual who is currently receiving aid under a different Aid Code. This scenario uses the data displayed

in Transaction 2. The following bullets detail the steps that the business logic follows to process a transaction that changes the Aid Code.

- ❑ The program will search for a row using the Key and Business Logic 3, and finds that row #2 contains the same identifying information
- ❑ The program then uses Business Logic 6, and inserts a new row (row #3)
- ❑ The program will set the overlap flag to Y, identifying that an overlap exists.

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>
CIN	123456789
Case Serial Number	1234567
FBU	1
Person Number	01
County Code	05
Program Effective Date	19980301
Program Type	04
Aid Code	3R
Federal/State Only Indicator	N
Program Status Code	Aid

Database Before Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>County Code</i>	<i>Aid Code</i>	<i>Program Type Code</i>	<i>Program Start Date</i>	<i>Program End Date</i>
1	123456789	1234567	1	01	05	90	19	19980101	Null
2	123456789	1234567	1	01	05	3H	04	19980101	Null

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>County Code</i>	<i>Aid Code</i>	<i>Program Type Code</i>	<i>Program Start Date</i>	<i>Program End Date</i>
1	123456789	1234567	1	01	05	90	19	19980101	Null
2	123456789	1234567	1	01	05	3H	04	19980101	Null
3	123456789	1234567	1	01	05	3R	04	19980301	Null

Scenario #3

This transaction will demonstrate how to terminate (end date) a program for a client who is receiving aid under a program for which the end date is not known (contains a null value).

- ❑ The program will follow the Key matching logic, Business Logic 3, and find a row that matches (row #3)

- ❑ The program then uses Business Logic 5, and because the record found contains a null Program End Date, and the Program Status Code for the data in Transaction 3 equals 'Disc', the Program End Date will be updated

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 3</i>
CIN	123456789
Case Serial Number	7654321
FBU	1
Person Number	01
County Code	05
Program Effective Date	19980530
Program Type	04
Aid Code	3R
Federal/State Only Indicator	N
Program Status Code	Disc

Database Before Image for Transaction 3

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>County Code</i>	<i>Aid Code</i>	<i>Program Type Code</i>	<i>Program Start Date</i>	<i>Program End Date</i>
1	123456789	7654321	1	01	05	90	19	19980101	Null
2	123456789	7654321	1	01	05	3H	04	19980101	Null
3	123456789	7654321	1	01	05	3R	04	19980301	Null

Database After Image for Transaction 3

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>County Code</i>	<i>Aid Code</i>	<i>Program Type Code</i>	<i>Program Start Date</i>	<i>Program End Date</i>
1	123456789	7654321	1	01	05	90	19	19980101	Null
2	123456789	7654321	1	01	05	3H	04	19980101	Null
3	123456789	7654321			05	3R	04	19980301	19980530

Transaction TRAC-Program Exceptions

Description

This transaction is designed to load data to the Program Exceptions table (SIS_PGM_EXCPT). This transaction uses similar logic to the one used for loading program participation data. It will use effective dates and status codes to build periods of exceptions for an individual with start and end dates that will be used by the program logic to do transaction operations such as inserting and updating records for an individual. The periods created by the transactions will then be used in the calculation of an individual's time clocks.

Business Logic

1. All data that is added to this table by a county will only affect other data for that county. The county where the data originated will always be the "owner" of the data
2. The fields used to compose the **Key** used to identify records in the database will vary depending on whether the county that is sending the data is a county that can run multiple Program Exceptions concurrently for an individual. If the county can run multiple Program Exceptions concurrently, the CIN, Case Serial Number, Person Number, FBU MEDS Code, Aid Code, Program Exception Reason Code, and County Code will compose the **Key**. If the county cannot run multiple Program Exceptions concurrently, the CIN, Program Exception Code and County Code will be used as the **Key** to identify records in the database
3. Through the use of the **Key** this program will reference the Program Exception Effective Date and Program Exception Status Code provided by the county to identify specific periods of eligibility, and update the data associated with those periods in the database
4. If the **Key**, and the Program Exception Effective Date do not locate a row where the Program Effective Date falls before the Program Exception End Date, and the Program Exception Status Code equals 'Aid' a new period will be inserted and the end date for the new exception will be defaulted to a value representing null
5. The Program Exception Status Code can end one period of eligibility (when Program Exception Status Code equals 'Disc') and begin another (when Program Exception Status Code equals 'Aid')
6. When the Program Exception Effective Date is added to the database as a Program Exception Start Date, all the other data received in the transaction will be added to the database in that same record
7. When the Program Exception Effective Date is added as a Program Exception End Date, no other fields will be updated for the existing row
8. The Program Exception Effective Month is an optional value that is used to synchronize cycle periods of eligibility with monthly periods of eligibility. This field is mandatory for Fresno County only. All other counties may leave this field blank
9. In some circumstances, the data that the county has sent may not be added because of the current condition of the data (i.e., a transaction with a Program Exception Status Code of 'Disc' has been sent with a Program Exception Date outside of all participation periods). These instances will be included in the Exception File sent back to the county

10. To correct a Program Exception Start Date for a period, a complete record containing the correct Program Exception Effective Date, the **Key**, and a Program Exception Status Code of 'Aid' needs to be sent. This Program Exception Effective Date will update the database as the new Program Exception Start Date
11. To correct a Program Exception End Date for a period, a complete record containing the correct Program Exception Effective Date, the **Key**, and a Program Exception Status Code of 'Disc' needs to be sent. This Program Exception Effective Date will update the database as the new Program Exception End Date
12. To delete an instance of an Exception for a period that has been sent because of an error, a transaction that contains the same Program Exception Effective Date, the **Key**, and a Program Exception Status Code of 'Disc' need to be sent
13. In order to send a complete program exception period (with Program Exception Start and Program Exception End Date) the county will need to send two separate transactions. The first transaction will set the Program Exception Start Date (with all associated fields), and the second transaction will set the Program Exception End Date

Scenarios

The following four scenarios have been identified to demonstrate, at a high level, the logic used to load exceptions related data into the TRAC database. All four scenarios assume that row #1, in the Database Image Table, already existed in the TRAC Application prior to the arrival of the transactions which are described in the following scenarios.

Scenario #1

This transaction will demonstrate how to insert a new record into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search for the Key following the logic described in Business Logic 2 and 3
- ❑ The search results did not find a row (there were no existing rows that matched the **Key** in the database). The program uses Business Logic 4 and inserts the row into the database with a Program Exception End Date equal to null

Transactions Received Table

Data Elements	Transaction 1
CIN	987654321
Case Serial Number	1234567
FBU	1
Person Number	01
Aid Code	30
County Code	34
Program Exception	19990101

<i>Data Elements</i>	<i>Transaction 1</i>
Effective Date	
Program Exception Code	03
Program Exception Reason Code	301
Program Exception Status Code	Aid

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430
2	987654321	1234567	1	01	34	03	301	19990101	Null

Scenario #2

This transaction will demonstrate how to make a correction to the transaction processed in Scenario #1. For this scenario, two transactions will be needed, the first one will discontinue the existing one, and the second one will insert the correct data in the database. This scenario uses the data displayed in Transaction 2 and 3. For this scenario, both transactions were received in the same load. The following bullets detail the steps that the business logic follows to process a transaction that changes the aid code.

- ❑ The program will first execute Transaction 2. It will search for a row using the **Key** and Business Logic 3, and finds that row #2 contains the same identifying information. The program uses Business Logic 11 and makes the correction to the row, which ultimately triggers the deletion of the row
- ❑ The program will then execute Transaction 3. Because a row is not found, it will use Business Logic 4 and insert the row into the database with a Program Exception End Date equal to null (row #3)

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>	<i>Transaction 3</i>
CIN	987654321	987654321
Case Serial Number	1234567	1234567

<i>Data Elements</i>	<i>Transaction 2</i>	<i>Transaction 3</i>
FBU	1	1
Person Number	01	01
Aid Code	30	30
County Code	34	34
Program Exception Effective Date	19990101	19990101
Program Exception Code	03	03
Program Exception Reason Code	301	303
Program Exception Status Code	Disc	Aid

Database Before Image for Transactions 2and 3

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430
2	987654321	1234567	1	01	34	03	301	19990101	Null

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430

Database After Image for Transaction 3

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430
3	987654321	1234567	1	01	34	03	303	19990101	Null

Scenario #3

This transaction will demonstrate how to change an end date for an exception.

- ❑ The program will follow the **Key** matching logic, Business Logic 3, and find a row that matches (row #3).
- ❑ The program then uses Business Logic 11, and because the row found contains a Program End Date, and the Program Exception Code for the data in Transaction 4 equals 'Disc', the Program Exception End Date will be updated.

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 4</i>
CIN	987654321
Case Serial Number	1234567
FBU	1
Person Number	01
Aid Code	30
County Code	34
Program Exception Effective Date	19990531
Program Exception Code	03
Program Exception Reason Code	303
Program Exception Status Code	Disc

Database Before Image for Transaction 4

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430
3	987654321	1234567	1	01	34	03	303	19990101	19990831

Database After Image for Transaction 4

<i>Row #</i>	<i>CIN</i>	<i>Case #</i>	<i>FBU</i>	<i>Person Number</i>	<i>Last Updated Entity</i>	<i>Program Exception Code</i>	<i>Program Exception Reason Code</i>	<i>Program Exception Start Date</i>	<i>Program Exception End Date</i>
1	987654321	1234567	1	01	34	03	300	19980101	19980430
3	987654321	1234567	1	01	34	03	303	19990101	19990531

Transaction TRAC-Welfare to Work

Description

This transaction is designed to load data to the Welfare to Work table (SIS_WTW). This transaction will allow counties to update existing welfare to work information for an individual.

Business Logic

1. All data that is added to this table by a county will only effect other data for that county. The county where the data originated will always be the “owner” of the data
2. The CIN and the County Code (obtained from the Record Header) will be used as a **Key** to search for matching records in the database
3. If a record is found in the database that matches the **Key** for the transaction received, the program will update the existing row with the new data that is received in the transaction. If a record is not found, the new row received will be added to the database
4. If the CalWORKs Extension Number is not provided, it will be assigned a default value that equals ‘0’
5. The CalWORKs Extension Number should only be sent when the CalWORKs 18 Month Indicator equals ‘18’. Valid values for CalWORKs Extension Number are ‘0-6’

Scenarios

The following two scenarios have been identified to demonstrate, at a high level, the logic used to load welfare to work related data into the TRAC database. The data received for both scenarios (Transactions 1 and 2) will be displayed, as well as the before and after images of the database once the transactions have been processed.

Scenario #1

This transaction will demonstrate how to update an existing record into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will use the Key using Business Logic 2 and 3
- ❑ Because a record already existed in the database, the existing record will be updated with the new data received in Transaction 1

Transactions Received Table

Data Elements	Transaction 1
CIN	123456789
County Code	07
Workplan Sign Date	19980501
CalWORKs 18 Month Indicator	18
CalWORKs Extension Number	0

<i>Data Elements</i>	<i>Transaction 1</i>
WTW Status	AID

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Workplan Sign Date</i>	<i>CalWORKs 18 Month Indicator</i>	<i>CalWORKs Extension Number</i>
1	123456789	03	19980313	18	0
2	123456789	07	19980301	18	0

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Workplan Sign Date</i>	<i>CalWORKs 18 Month Indicator</i>	<i>CalWORKs Extension Number</i>
1	123456789	03	19980313	18	0
2	123456789	07	19980501	18	0

Scenario #2

This transaction will demonstrate how to insert a record for a client who up until the arrival of Transaction 2 was not known to the TRAC database. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ☐ The program will use the Key using Business Logic 2 and 3
- ☐ Because no records are found the new row of data is added to the table

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>
CIN	111111111
County Code	07
Workplan Sign Date	19980701
CalWORKs 18 Month Indicator	18
CalWORKs Extension Number	0
WTW Status	Aid

Database Before Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Workplan Sign Date</i>	<i>CalWORKs 18 Month Indicator</i>	<i>CalWORKs Extension Number</i>
1	123456789	03	19980313	18	0
2	123456789	07	19980501	18	0

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Workplan Sign Date</i>	<i>CalWORKs 18 Month Indicator</i>	<i>CalWORKs Extension Number</i>
1	123456789	03	19980313	18	0
2	123456789	07	19980501	18	0
3	111111111	07	19980701	18	0

Transaction TRAC-Non-California Months Participation

Description

This transaction is designed to load data to the Non-California Participation table (SIS_NONCAPT). This transaction uses similar logic to the one used for loading program participation data. It will use effective dates and status codes to build periods of participation for an individual with start and end dates that will be used by the program logic to do transaction operations such as inserting and updating records for an individual. The periods created by the transactions will then be used in the calculation of an individual's time clocks.

Business Logic

1. All data that is added to this table by a county will only effect other data for that county. The county where the data originated will always be the "owner" of the data
2. The CIN, Non-California State Code, and County Code (obtained from the Record Header) will be used as a **Key** to identify records in the database
3. Through the use of the **Key** this program will reference the Non-California Participation Effective Date and Non-California Status Code provided by the county to identify specific periods of eligibility, and update the data associated with those periods in the database
4. If the **Key** does not locate a row where the Non-California Participation Effective Date falls within the period of the Non-California Start Date and Non-California End Date, a new period will be inserted and the end date for the new program will be defaulted to a value representing null
5. If the **Key** locates a row where the key is the same, the Non-California end Date contains a value representing null, and the Non-California Status Code for the transaction record equals 'Aid'. The program will end the existing period of participation by subtracting a day from the Non-California Participation Effective Date received in the new transaction, and insert the new row with the Non-California Participation Effective Date as the new record's Non-California Start Date
6. Based on the Non-California Status Code, the data can end one period of participation, when Non-California Status Code equals 'Disc', and begin another, when Non-California Status Code equals 'Aid' to indicate a change in Non-California Status Code
7. When the Non-California Participation Effective Date is added as a Non-California Start Date, all the data associated with the Non-California Participation Effective Date (from the county) will be added to the database in that same record
8. In some circumstances, the data that the county has sent may not be added because of the current condition of the data (i.e., a transaction with a Non-California Status Code of 'Disc' has been sent with a Non-California Participation Effective Date outside of all participation periods). These instances will be included in the Exception File sent back to the county
9. To correct a Non-California Start Date for a period that has a Non-California Start Date, a transaction must be sent with a Non-California Status Code of 'Aid', the correct Non-California Participation Effective Date, and the **Key**. Since this Non-California Participation Effective Date will populate the database as a Non-California Start Date

10. Counties must send a complete non-California program participation period (with Program Start and Program End Dates) consisting of two separate transactions. The first transaction will set the Non-California Start Date (with all associated fields), and the second transaction will set the Non-California End Date

Scenarios

The following scenario has been identified to demonstrate, at a high level, the logic use to load Non-California program participation related data into the TRAC database. The following three scenarios assume that row #1, in the Database Image Table, already existed in the TRAC Application prior to the arrival of the transactions which are described in the following scenarios. For more examples of the process used to load Non-California Program Participation information, please refer to the scenarios in the Program Participation Transaction and Program Exception transactions subsections of the document, as the logic used to load data in these transactions is the same as the one used to load Non-California data.

Scenario #1

This transaction will demonstrate how to insert a new complete record (contains start and end periods) into the TRAC database. This scenario will be accomplished through the data contained in Transactions 1 and 2 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database

- ❑ The program will execute Transaction 1 and search for the Key following the logic described in Business Logic 3
- ❑ Because the Key did not find a row that matched, the program uses Business Logic 4 and inserts the new row into the database with a Program End Date equal to null
- ❑ The program will then execute Transaction 2 and search for the Key following the logic described in Business Logic 3
- ❑ Because the Key found a row that matched, the program uses Business Logic 6 and inserts the new Program End Date. These two transactions have demonstrated how data needs to be sent in order to insert one full period of Non-California Program Participation

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 1</i>	<i>Transaction 2</i>
CIN	111111111	111111111
Non-California State Code	MN	MN
County Code	01	01
Non-California Participation Effective Date	19980301	19980930
Non-California Status Code	Aid	Disc

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Non-California State Code</i>	<i>Non-Cal Start Date</i>	<i>Non-Cal End Date</i>	<i>Last Updated Entity (stores the County Code)</i>
1	111111111	MN	19960101	19960130	01

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Non-California State Code</i>	<i>Non-Cal Start Date</i>	<i>Non-Cal End Date</i>	<i>Last Updated Entity (stores the County Code)</i>
1	111111111	MN	19960101	19960130	01
2	111111111	MN	19980301	Null	01

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Non-California State Code</i>	<i>Non-Cal Start Date</i>	<i>Non-Cal End Date</i>	<i>Last Updated Entity (stores the County Code)</i>
1	111111111	MN	19960101	19960130	01
2	111111111	MN	19980301	19980930	01

Transaction TRAC-Diversion Program Participation

Description

This transaction is designed to load data to the Diversion participation table (SIS_PGM_DIV). The business logic developed for this program is based on the concept of adding and deleting rows according to the status code (Diversion Status Code) that is received in the transaction. If the status code of the row received equals 'Aid' the row will be added or updated. To make corrections to the existing data (delete a Diversion Period), a transaction with a status code of 'Disc' will need to be sent.

Business Logic

1. All data that is added to this table by a county will only effect other data for that county. The county where the data originated will always be the "owner" of the data
2. The CIN, Diversion Payment Date, and County Code (obtained from the Record Header) will be used as a **Key** to identify records in the database
3. The **Key** will be used to match records within the database. If the record does not match the **Key** the new data received in the transaction will be added as a new row
4. If the **Key** finds a row in the database, and the Diversion Status Code of the new record equals 'Aid', the received row will update the existing row
5. If the **Key** finds a row in the database, and the Diversion Status Code of the new record equals 'Disc', the existing row in the database will be physically deleted from the database

Scenarios

The following two scenarios have been identified to demonstrate, at a high level, the logic used to load Diversion Program Participation data into the TRAC database. The data received for the two scenarios (Transactions 1 and 2) will be displayed, as well as the before and after images of the database once the transactions have been processed.

Note: Multiple Diversion Payments within a month will only count as one month in the welfare time clocks, regardless of whether the Diversion Aid Code of those payments is different.

Scenario #1

This transaction will demonstrate how to insert a month of Diversion only into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 3, because the row is not found, the new data received in Transaction 1 will be inserted as a new row in the table

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 1</i>
CIN	000000000
Payment Date	19980512
County Code	09
Diversion Status Code	Aid
Diversion Start Month	199805
Diversion End Month	199807

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Diversion Payment Date</i>	<i>Diversion Start Month</i>	<i>Diversion End Month</i>
1	000000000	03	19970501	199705	199707

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Diversion Payment Date</i>	<i>Diversion Start Month</i>	<i>Diversion End Month</i>
1	000000000	03	19970501	199705	199707
2	000000000	09	19980512	199805	199807

Scenario #2

This transaction will demonstrate how to make a correction for a Diversion Payment into the TRAC database for an individual. This scenario will be accomplished through the data contained in Transaction 2 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 5. Because a row is found, the program will look at the Diversion Status Code, and since the code received is a 'Disc' code, the existing database row that is matched will be deleted

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>
CIN	000000000
Payment Date	19970501
County Code	03
Diversion Status Code	Disc

Database Before Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Diversion Payment Date</i>	<i>Diversion Start Month</i>	<i>Diversion End Month</i>
1	000000000	03	19970501	199705	199707
2	000000000	09	19980512	199805	199807

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Diversion Payment Date</i>	<i>Diversion Start Month</i>	<i>Diversion End Month</i>
2	000000000	09	19980512	199805	199807

Transaction TRAC-Supportive Services Only

Description

This transaction is designed to load data to the Supportive Services Only table (SIS_SUP_SRV). The business logic developed for this program is based on the concept of adding and deleting individual instances of Supportive Services Only aid according to the status code (Supportive Services Status Code) that is received in the transaction. Because Supportive Services Only aid is granted for a period of one month, if the status code equals 'Aid' the row will be added. To make corrections to the existing data (delete a Supportive Services Period), a transaction with a status code of 'Disc' will need to be sent.

Business Logic

1. All data that is added to this table by a county will only effect other data for that county. The county where the data originated will always be the "owner" of the data
2. The CIN, Supportive Services Reason Code, Supportive Services Effective Month, and County Code (obtained from the Record Header) will be used as a **Key** to identify records in the database
3. The **Key** will be used to match records within the database. If the record does not match the **Key**, the new data received in the transaction will be added as a new row
4. If the **Key** finds a match in the database, and the Supportive Services Program Status Code of the new record equals 'Aid', the received row will update the record in the database
5. If the **Key** finds a row in the database, and the Supportive Services Program Status Code of the new record equals 'Disc', the existing row in the database will be physically deleted from the database

Scenarios

The following three scenarios have been identified to demonstrate, at a high level, the logic used to load Supportive Services Only data into the TRAC database. The data received for the three scenarios (Transactions 1, 2, and 3), as well as the before and after images of the database will be displayed once the transactions have been processed.

Scenario #1

This transaction will demonstrate how to insert a month of Supportive Services Only into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 3. Because the row is not found, the new data received in Transaction 1 will be inserted as a new row in the table

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 1</i>
CIN	222222222
Supportive Services Reason Code	01
Supportive Services Effective Month	199907
County Code	09
Supportive Services Status Code	Aid

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Supportive Services Reason Code</i>	<i>Supportive Services Effective Month</i>
1	222222222	03	01	199905

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Supportive Services Reason Code</i>	<i>Supportive Services Effective Month</i>
1	222222222	03	01	199905
2	222222222	09	01	199907

Scenario #2

This transaction will demonstrate how to insert a month of Supportive Services Only information into the TRAC database for an individual who already has had Supportive Services aid granted for that month. This scenario will be accomplished through the data contained in Transaction 2 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 3. Even though the individual is matched, the whole Key is not matched because the Supportive Service Program Type Code is different. Therefore, the new data received in Transaction 2 will be inserted as a new row in the table

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>
CIN	222222222
Supportive Services	03

<i>Data Elements</i>	<i>Transaction 2</i>
Reason Code	
Supportive Services Effective Month	199907
County Code	09
Supportive Services Status Code	Aid

Database Before Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Supportive Services Reason Code</i>	<i>Supportive Services Effective Month</i>
1	22222222	03	01	199905
2	22222222	09	01	199907

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Supportive Services Reason Code</i>	<i>Supportive Services Effective Month</i>
1	22222222	03	01	199905
2	22222222	09	01	199907
3	22222222	09	03	199907

Scenario #3

This transaction will demonstrate how to make a correction for Supportive Services Only information into the TRAC database for an individual. This scenario will be accomplished through the data contained in Transaction 3 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 5. Because a row is found, the program will look at the Supportive Service Only Status Code, and since the code received is a 'Disc' code, the existing database row that is matched will be deleted

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 3</i>
CIN	22222222
Supportive Services Reason Code	01
Supportive Services Effective Month	199907
County Code	09
Supportive Services Status Code	Disc

Database Before Image for Transaction 3

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Supportive Services Reason Code</i>	<i>Supportive Services Effective Month</i>
1	222222222	03	01	199905
2	222222222	09	01	199907
3	222222222	09	03	199907

Database After Image for Transaction 3

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Supportive Services Reason Code</i>	<i>Supportive Services Effective Month</i>
1	222222222	03	01	199905
3	222222222	09	03	199907

Transaction Child Support Reimbursement

Description

This transaction is designed to load data to the Child Support Reimbursement table (SIS_CHLD_SUP_REIMB). The business logic is based on the same concept as the one developed for Supportive Services Only. This program will add and delete individual instances of Child Support Reimbursement according to the status code (Child Support Status Code) that is received in the transaction. Because Child Support Reimbursement is granted for a period of one month, if the status code equals 'Aid' the row will be added. To make corrections to the existing data (delete a Child Support Effective Month), a transaction with a status code of 'Disc' will need to be sent.

Business Logic

1. All data that is added to this table by a county will only effect other data for that county. The county where the data originated will always be the "owner" of the data
2. The CIN, Child Support Effective Month, and County Code (obtained from the Record Header) will be used as a **Key** to identify records in the database
3. The **Key** will be used to match records within the database. If the record does not match the **Key** the new data received in the transaction will be added as a new row
4. If the **Key** finds a row in the database, and the Child Support Status Code of the new record equals 'Aid', the received row will be rejected because the received row contains the same information as the row that exists in the database
5. If the **Key** finds a row in the database, and the Child Support Status Code of the new record equals 'Disc', the existing row in the database will be physically deleted from the database

Scenarios

The following two scenarios have been identified to demonstrate, at a high level, the logic used to load Supportive Services Only data into the TRAC database. The data received for two scenarios (Transactions 1 and 2), as well as the before and after images of the database will be displayed once the transactions have been processed.

Scenario #1

This transaction will demonstrate how to insert a month of Child Support Reimbursement into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 3. Because the row is not matched, the new data received in Transaction 1 will be inserted as a new row in the table

Transactions Received Table

Data Elements	Transaction 1
CIN	333333333

<i>Data Elements</i>	<i>Transaction 1</i>
Child Support Effective Month	199902
County Code	23
Child Support Reimbursement Status Code	Aid

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Child Support Effective Month</i>
1	333333333	05	199805

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Child Support Effective Month</i>
1	333333333	05	199805
2	333333333	23	199902

Scenario #2

This transaction will demonstrate how to make a correction for Child Support Reimbursement in the TRAC database for an individual. This scenario will be accomplished through the data contained in Transaction 2 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the Key using Business Logic 5. Because a row is found, the program will look at the Child Support Reimbursement Code, and since the code received is a 'Disc' code, the existing database row that is matched will be deleted

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>
CIN	333333333
Child Support Effective Month	199805
County Code	05
Child Support Reimbursement Status Code	Disc

Database Before Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Child Support Effective Month</i>
--------------	------------	---	--------------------------------------

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Child Support Effective Month</i>
1	333333333	05	199805
2	333333333	23	199902

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Child Support Effective Month</i>
2	333333333	23	199902

Transaction TRAC-Under 10 Grant

Description

This transaction is designed to load data regarding Grants Under 10 dollars (i.e.: zero basic grants) in the related TRAC database table (SIS_UNDER10_GRANT). The business logic is based on the same concept as the one developed for Supportive Services Only and Child Support Reimbursement. This program will add and delete individual instances of grant payments according to the status code (Under 10 Grant Status Code) that are received in the transaction. Because these type of grants are granted for a period of one month, if the status code equals 'Aid' the row will be added. To make corrections to the existing data (delete an Under 10 Grant), a transaction with a status code of 'Disc' will need to be sent.

Business Logic

1. All data that is added to this table by a county will only effect other data for that county. The county where the data originated will always be the "owner" of the data
2. The CIN, Under 10 Grant Effective Month, and County Code (obtained from the Record Header) will be used as a **Key** to identify records in the database
3. The **Key** will be used to match records within the database. If the record does not match the **Key** the new data received in the transaction will be added as a new row
4. If the **Key** finds a row in the database, and the Under 10 Grant Status Code of the new record equals 'Aid', the received row will be rejected because the received row contains the same information as the row that exists in the database
5. If the **Key** finds a row in the database, and the Under 10 Grant Code of the new record equals 'Disc', the existing row in the database will be physically deleted from the database

Scenarios

The following two scenarios have been identified to demonstrate, at a high level, the logic used to load Grants Under 10 dollars data into the TRAC database. The data received for two scenarios (Transactions 1 and 2) will be displayed, as well as the before and after images of the database once the transactions have been processed.

Scenario #1

This transaction will demonstrate how to insert a month, in which the client received a payment of a grant that is under 10 dollars, into the TRAC database. This scenario will be accomplished through the data contained in Transaction 1 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the **Key** using Business Logic 3, because the row is not matched, the new data received in Transaction 1 will be inserted as a new row in the table

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 1</i>
CIN	444444444
Under 10 Grant Effective Month	199902
County Code	23
Under 10 Grant Status Code	Aid

Database Before Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Under 10 Grant Effective Month</i>
1	444444444	27	199807

Database After Image for Transaction 1

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Under 10 Grant Effective Month</i>
1	444444444	27	199807
2	444444444	23	199902

Scenario #2

This transaction will demonstrate how to make a correction for a payment of a Grant that is under 10 dollars in the TRAC database. This scenario will be accomplished through the data contained in Transaction 2 in the Transactions Received Table. The following bullets detail the steps that the business logic follows in order to insert the new record into the TRAC database.

- ❑ The program will search on the **Key** using Business Logic 5. Because a row is found, the program will look at the Under 10 Grant Status Code, and since the code received is a 'Disc' code, the existing database row that is matched will be deleted

Transactions Received Table

<i>Data Elements</i>	<i>Transaction 2</i>
CIN	444444444
Under 10 Grant Effective Month	199902
County Code	23
Under 10 Grant Status Code	Disc

Database Before Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Under 10 Grant Effective Month</i>
--------------	------------	---	---------------------------------------

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Under 10 Grant Effective Month</i>
1	444444444	27	199807
2	444444444	23	199902

Database After Image for Transaction 2

<i>Row #</i>	<i>CIN</i>	<i>Last Updated Entity (stores the County Code)</i>	<i>Under 10 Grant Effective Month</i>
1	444444444	27	199807

Transmission Protocols

Data transmissions between the counties and the TRAC Application will be based on the process currently in use for transmissions between the county and the MEDS application. The current data transmission process utilizes several functions of the mainframe-operating environment. Definitions for those functions are provided to assist in better understanding the rest of this section. The following terms will be used in this section:

Internal Reader - The internal reader (INTRDR) function accepts JES3 output data as an input job stream. Any job can use the internal reader to pass a job stream to JES3. The internal reader allows jobs submitted from MVS/TSO/E SUBMIT command and jobs created as system output (SYSOUT=(class, INTRDR)) to be submitted to JES3 input service for processing. JES3 dynamically starts internal readers as needed.

JES – The Job Entry System is a function within the operating system of mainframe computers that facilitates the execution of tasks, called JOBS, on the computer. The multiple versions of JES are JES2 and JES3. JES3 is in use at the HHSDC.

TCP/IP –Transmission Control Protocol/Internet Protocol (TCP/IP) is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in the private networks called intranets and extranets.

XMIT – Transmit (shortened to XMIT for JES3 command syntax) is a mainframe function that allows files and information to be sent from one location to another using the TCP/IP network.

The current MEDS data file transmission uses the XMIT and Internal Reader facilities of the JES3 environment to send files of data between the county computing environment and the HHSDC computing systems. The following subsections will describe the process in further detail.

Data Transmissions from County to TRAC

Data will be transmitted to the TRAC Application on a daily schedule. The file transfer process will include the following steps:

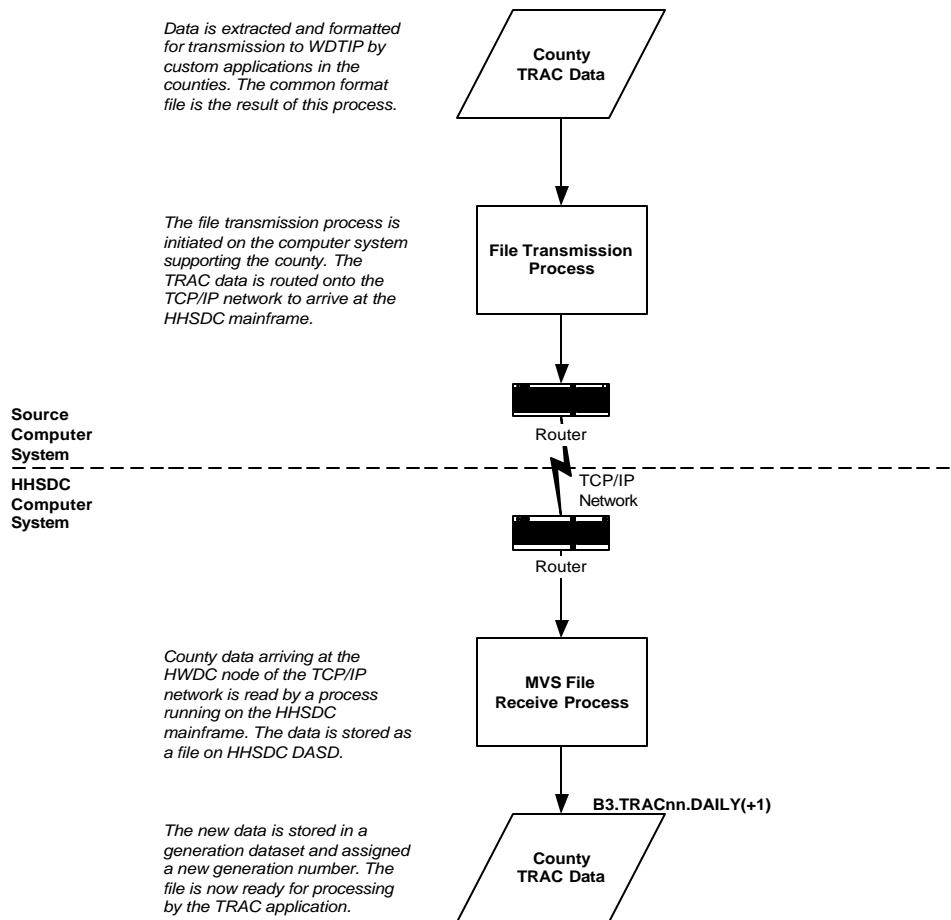
- ❑ *County Data Extraction File:* Data is extracted from the county database(s) and formatted into the extraction file layout described in the ***Developer Notes*** section of this document
- ❑ *File Transmission:* The extraction file is input to a transmission function on the county computer system. This function contains the routing code for the HHSDC and the output file name for TRAC Application daily extraction. File names are defined by the TRAC Application and will be in the format B3.TRACnn.DAILY(+1). TRACnn will be changed to reflect the county number of the originating county, thus Sacramento would send a file named B3.TRAC34.DAILY(+1). The (+1) parameter refers to the creation of a new generation of a defined generation dataset group. Generation datasets will be defined by the TRAC Application to receive the county data

- ❑ **File Reception:** Data arriving at the HHSDC node of the TCP/IP network will be read by a process running on the HHSDC mainframe. The data will be cataloged into the files named in the transmission data

At the completion of these processes, the county data will be cataloged and stored on DASD associated with the HHSDC mainframe. At that time it will be available for processing by the application

The following figure depicts these activities and shows the flow of data between the source systems and the HHSDC.

Figure 14: County to TRAC File Transfer



Data Transmissions from the TRAC Application to County

Data will be transmitted to the counties on a scheduled basis. The file transfer process will include the following steps:

- ❑ *The TRAC Application File Creation:* Data resulting from TRAC's periodic processing will be stored in files on the HHSDC. These files may contain a variety of information such as reports or errors encountered in the processing of county data
- ❑ *File Transmission:* The files are input to a transmission function on the HHSDC. This function contains the routing code for the county and the output file name(s). File names are defined by the receiving county
- ❑ *File Reception:* Data arriving at the county node of the TCP/IP network will be read by a process running on the county computer. The data will be saved into the files named in the transmission data

At the completion of these processes, the county data will be stored on media associated with the county computer. At that time it will be available for processing or review by the county.

Transmission Methods in Use by Counties

Two methods of file transmission are in use. Counties with independent processing facilities may send their daily information directly to the HHSDC for processing by the MEDS application. Counties that are members of the ISAWS consortia have their data processed by the ISAWS data center. The following table shows the transmission protocol currently in use by each of the counties. The processes are described further in the items following the chart.

<i>County Number and Name</i>		<i>Consortia</i>	<i>Independent Host to Host Transmission</i>	<i>Combined File Transmission</i>
1	Alameda	C-III	X	
2	Alpine	C-I		X
3	Amador	C-I		X
4	Butte	C-I		X
5	Calaveras	C-I		X
6	Colusa	C-I		X
7	Contra Costa	C-III	X	
8	Del Norte	C-I		X
9	El Dorado	C-I		X
10	Fresno	C-III	X	
11	Glenn	C-I		X
12	Humboldt	C-I		X
13	Inperial	C-I		X
14	Inyo	C-I		X
15	Kern	C-I		X
16	Kings	C-I		X
17	Lake	C-I		X
18	Lassen	C-I		X
19	Los Angeles	C-II	X	
20	Madera	C-I		X
21	Marin	C-I		X
22	Mariposa	C-I		X

<i>County Number and Name</i>		<i>Consortia</i>	<i>Independent Host to Host Transmission</i>	<i>Combined File Transmission</i>
23	Mendocino	C-I		X
24	Merced	C-IV	X	X
25	Modoc	C-I		X
26	Mono	C-I		X
27	Monterey	C-I		X
28	Napa	C-I		X
29	Nevada	C-I		X
30	Orange	C-III	X	
31	Placer	C-III	X	
32	Plumas	C-I		X
33	Riverside	C-IV	X	
34	Sacramento	C-III	X	
35	San Benito	C-I		X
36	San Bernardino	C-IV	X	
37	San Diego	C-III	X	
38	San Francisco	C-III	X	
39	San Joaquin	C-I		X
40	San Luis Obispo	C-III	X	
41	San Mateo	C-III	X	
42	Santa Barbara	C-III	X	
43	Santa Clara	C-III	X	
44	Santa Cruz	C-III	X	
45	Shasta	C-I		X
46	Sierra	C-I		X
47	Siskiyou	C-I		X
48	Solano	C-III	X	
49	Sonoma	C-III	X	
50	Stanislaus	C-IV	X	
51	Sutter	C-I		X
52	Tehama	C-I		X
53	Trinity	C-I		X
54	Tulare	C-III	X	
55	Tuolumne	C-I		X
56	Ventura	C-III	X	
57	Yolo	C-III	X	
58	Yuba	C-I		X

- ❑ *Independent Host to Host Transmission:* Data is input to a JES3 batch job that submits a transmit function to the internal reader. The XMIT function will execute and route to the JES queue of the HHSDC. County IT analysts may model the new send functions on the existing MEDS send function. Transmit and internal reader parameters will not need to be modified in order to create a TRAC Application send function
- ❑ *Combined File Transmission:* The counties that are part of the ISAWS consortia do not perform a direct data transmission from their county computer system to the HHSDC. Data is sent from the computer supporting the ISAWS system to the HHSDC MEDS application on behalf of the member counties. The process currently in use invokes the File Transfer Processing (FTP) function to send a combined daily file to the Internet Protocol (IP) address of the HHSDC. The process is initiated on the UNISYS 2200 mainframe that supports the ISAWS consortium

County IT analysts may model the new send functions on the existing MEDS send function. IP address parameters will not need to be modified in order to create a TRAC Application send function. New input and output file names and IBM site information will be required. The output file information will be supplied by the TRAC Application and will include record size, blocking, space, and retention period information.

Exception File Overview

This section explains the exception handling process that will be implemented in the TRAC Application to ensure that received county data is correct and can be loaded into the database. This section also explains the Exception File Layout and provides scenarios explaining the usage of the Exception File.

Process Overview

Exception handling will be a very critical process that will ensure that the data that passes the system data validation rules is loaded into the TRAC database. This process will be especially important in the TRAC Application because the majority of all inserts and updates to the database will be done through county extraction files sent on a daily basis. As each county's extraction file is processed, all of the data that fails to meet the validation requirements will be written to an exception file. The TRAC Application will reject data at two levels; file level and transaction level. The exception files generated will contain all of the rejected data for a specific county, as well as the reasons for rejection. Exception files will be sent on a daily basis to the system of origin so that the data can be corrected and resent to the TRAC Application.

File Level Rejection

This type of rejection will occur when the data sent in the Batch Header and/or Batch Trailer Record of the Extraction File fails to meet the data validation criteria of the TRAC Application. If a file level rejection occurs, none of the data for the file will be processed. To correct a file level rejection, counties must resend the entire file with the correct Batch Header and Batch Trailer Record information. Also, to indicate that the file is being sent a second time, the Transaction Version field in the Batch Header, Batch Trailer Record and Record Header must be increased by one. If the file is sent more than once the Transaction Version field must be sequentially incremented based on the number of times that the file has been sent.

Transaction Level Rejection

This type of rejection will take place when any of the data sent as part of the Transaction data and/or the Record Header fails to meet the data validation criteria of the TRAC Application. If a Transaction level rejection occurs, none of the data contained in that Transaction will be processed. Once the data has been fixed, the correction Transaction can be included as part of another day's Extraction File. When a Transaction level rejection occurs due to bad or missing data being sent as part of the Record Header, all of the transactions for that specific header will be rejected. If the number of rejected rows is substantial, counties may choose to send all of the corrected rows as part of a different version of the previous extraction file. The correction file would need to include the same Header File information, with the exception of a Transaction Version that has been increased according to the version, and containing only the rows that need to be corrected.

Data Correction Process

Counties are responsible for processing Exception Files, correcting the rejected data, and sending the valid data to the TRAC Application. The exception handling process is critical to ensure that the application is loaded with valid data only; however, for the process to be effective, counties must take responsibility for the data that is sent in their extraction files. The data sent should be a replica of the transactions that took place at the county system for a specific day so that the TRAC database contains an image of that data. In order for the TRAC Application to accurately calculate the three welfare time clocks, counties need to actively participate in the data loading process. When exceptions have occurred and data correction is needed, counties must be prompt in fixing the data in their systems and sending the correct data to the TRAC Application to prevent a mismatch between both systems. Any mismatch could result in an inaccurate calculation of welfare time clocks.

Exception File Layout

The exception file layout is designed similar to the extraction file provided by the counties, and has three major sections:

- ❑ Batch Header
- ❑ Batch Trailer Record
- ❑ Extraction File Transaction
- ❑ Exception Record Detail

Batch Header

The Batch Header contains information that identifies the exception report, and extraction file from which it was generated. The specific values that are contained in the Batch Header (and their associated description) include the following:

<i>Field Name</i>	<i>Description</i>
Transaction Code	Identifies the file as an exception file. The value "EXCP" should always populate this field.
Transaction Version	The version of the extraction file sent from the county. This field along with the Batch Number and County Code uniquely identify from which this file was generated.
County Code	Identifies the county of origin. It is used to uniquely identify from which this exception file was generated. A sample value for this field is 19 (for Los Angeles).
Batch Number	The sequential number generated by the county for the extraction file.

Batch Trailer Record

The Batch Trailer Record contains the same information the Batch Header, except for the following fields:

<i>Field Name</i>	<i>Description</i>
Transaction Code	Identifies the end of the file. The value "TRAL" should always populate this field.
Total Number of Records Processed	The number of records that successfully updated into the TRAC database. If there was an error on the Batch Header information from the extraction file, this value will be "0".
Total Number of Records	The total number of records provided in the extraction file. This field will directly be populated with the data in the Batch Trailer Record in the extraction file.

Extraction File Transaction

The extraction file transaction will be the complete record where the error occurred in the Extraction file. Since each extraction file transaction has a slightly different layout (see **Attachment A- Extraction File Layout**), this section of the exception file will also have a different structure based on the original transaction. Directly following the extraction file transaction record will be details on why the specific records was excepted. This data will be populated in the exception record detail section of the extraction file.

Exception Record Detail

The exception record detail contains a copy of the rejected transaction and the reason it was not processed. The Exception Record Detail contains the following values:

<i>Field Name</i>	<i>Description</i>
Extraction File Transaction	This area will be the exact record rejected from the Extraction File sent by the county. Since each transaction type in that file may have a different length, the remained of this field will be filler.
Reason Code	A three-digit code that describes the reason the record was rejected.
Reason Code Description	A textual description of the Reason Code. This field gives a reason the record was rejected in English.

Exception File Scenarios

In order to provide a clear understanding of the process by which data will be returned to the counties via the exception file, a few scenarios have been included in this document. These scenarios will describe the layout of the file and provide the reason/s data may be rejected by the load program. All of the exception file records described below will be found in the Extraction File Transaction and the Exception Record Detail of the exception file.

Scenario A- File Level Information Rejection

San Bernardino sends a file that does not contain a valid Account Code (it was a null value). Since this data element is mandatory and exists in the Batch Header and Batch Trailer Record, two records will be returned in the exception file.

Batch Header

<i>Field</i>	<i>Data</i>
Transaction Code	EXTR

<i>Field</i>	<i>Data</i>
Transaction Version	01
County Code	36
Account Code	(NULL)
Creation Date	20000724
Creation Time	120458
Batch Number	0012

Batch Trailer Record

<i>Field</i>	<i>Data</i>
Transaction Code	TRAL
Transaction Version	01
County Code	36
Account Code	(NULL)
Creation Date	20000724
Creation Time	120458
Batch Number	0012
Total Number of Records	0000005045

The records sent to San Bernardino in the exception record detail will look similar to the table below (the heading has been changed for display in this document):

Exception Detail Record

<i>Extraction File Transaction</i>	<i>RSN Code</i>	<i>Reason Code Description</i>
(An exact copy of the record that was rejected from the Extraction File followed by filler space if necessary)	103	Account Code is not a valid value or is null.
(An exact copy of the record that was rejected from the Extraction File followed by filler space if necessary)	113	Account Code is not a valid value or is null.

Scenario B - Record Header Information Missing

Kern County sends two transactions, Program Participation and Program Exception, for an individual that do not contain a value for Birth Date. The Birth Date is a mandatory element on the Record Header, so all records that are missing that element will be rejected from the loading process and included in the exception file. For the purpose of this example, the data in the record header would be duplicated, except for the specific Transaction Code for Program Exception (LD09).

Record Header (Program Participation)

<i>Field</i>	<i>Data</i>
Transaction Code	LD02
Transaction Version	01

<i>Field</i>	<i>Data</i>
County Code	15
Creation Date	20000801
Creation Time	080446
Batch Number	0024
CIN ID	123456789
Case Serial Number	7654321
Case FBU MEDS Code	(Null)
Person Number	(Null)
Aid Code	(Null)
Last name	(Null)
First Name	(Null)
Middle Name	(Null)
SSN	(Null)
Birth Date	(Null)
Gender	(Null)
Caseworker ID	00000EW345

The other relevant data for this example is displayed in the tables below.

Program Participation Transaction

<i>Field</i>	<i>Data</i>
Record Number	1199
Program Type Code	01
Aid Code	30
Federal/State Only Indicator	N
Program Effective Date	19991201
Program Effective Month	(Null)
Participation Type Code	A
Minor Parent Flag	N
Case Serial Number	7654321
Case FBU MEDS Code	(Null)
Person Number	(Null)
Program Discontinuance Code	(Null)
Program Status Code	Aid

Program Exception Transaction

<i>Field</i>	<i>Data</i>
Record Number	1204
Program Exception Code	03

<i>Field</i>	<i>Data</i>
Program Exception Reason Type Code	304
Program Exception Effective Date	20001201
Program Exception Effective Month	(Null)
Program Exception Status Code	Aid

Since the missing element appears in two different transactions in the extraction file, Kern will receive two different records for this data error. Since the Record Header information would be duplicated in this scenario for each record, it has been omitted.

Exception Detail Record

<i>Extraction File Transaction</i>	<i>RSN Code</i>	<i>Reason Code Description</i>
(An exact copy of the record that was rejected from the Extraction File followed by filler space if necessary)	135	Birth Date is not a valid value or is null.
(An exact copy of the record that was rejected from the Extraction File followed by filler space if necessary)	135	Birth Date is not a valid value or is null.

In this case, a birth date would need to be added and then the records would be resent as a component of the next daily load (where the Batch Number would theoretically increment by one) or in a separate transaction which would include only data corrections from this original file (in which the Transaction Version Number would be incremented).

It is important to note that the entire record would be rejected for both the Program Participation Transaction and the Program Exception Transaction and none of the associated information for those transactions would be loaded into the TRAC Application.

Scenario C – Field Level Information Missing

If data is passed to the TRAC Application that is invalid or a mandatory field is missing, the entire transaction will be rejected. For example, Orange County sends a transaction for an individual to update the Welfare to Work information for an individual.

Welfare to Work Transaction

<i>Field</i>	<i>Data</i>
Record Number	65
Workplan Sign Date	19990126
CalWORKs 18 Month Indicator	18
CalWORKs Extension Number	No

Since the valid values for the CalWORKs Extension Number is 0-6, “No” is invalid. For this reason the entire record would be rejected.. Again the Record Header information is not displayed in this scenario and the Extraction File Transaction is represented by the Transaction Code for Welfare to Work Information (LD03).

Exception Detail Record

<i>Extraction File Transaction</i>	<i>RSN Code</i>	<i>Reason Code Description</i>
(An exact copy of the record that was rejected from the Extraction File followed by filler space if necessary)	172	CalWORKs Extension Number is not a valid value.

Scenario D – Warning Messages for Data Loads

As discussed in the Extraction File Layout sub-section of this document, if two records are received with data that seems to be problematic but is not invalid, the TRAC Application will load the data and send a warning message back in the Extraction File. This can occur in three types of transactions:

- ☐ Program Participation
- ☐ Program Exceptions
- ☐ Non-California Program Participation

One example of this warning would be when a county sends a transaction that does not change any of the data for an existing record. For example, Los Angeles sends a Program Exception record to the TRAC Application that has the same Program Exception Code and Program Exception Reason Code as the previous record in the database. Additionally, the record in the database does not have an end date, and the new record has a Program Exception Status Code of “Aid” (meaning it is not designed to only end date the current record). Again, only the Transaction Code has been included in the example, but the actual file would contain the entire record including the Record Header and Transaction. Also, Los Angeles does not allow for concurrent program exception (in their source system) in this example.

Program Exception Transaction

<i>Field</i>	<i>Data</i>
Record Number	20736
Program Exception Code	03
Program Exception Reason Code	304
Program Exception Effective Date	19990101
Program Exception Effective Month	(Null)

<i>Field</i>	<i>Data</i>
Program Exception Status Code	Aid

This is how the exception report will return a message to the county.

Exception Detail Record

<i>Extraction File Transaction</i>	<i>RSN Code</i>	<i>Reason Code Description</i>
(An exact copy of the record that was rejected from the Extraction File followed by filler space if necessary)	301	Warning- two consecutive Program Exception records were received with a Program Exception Status Code = 'Aid' and no other differences in the records. THIS RECORD WAS LOADED TO THE TRAC DATABASE.

The Reason Code for all warning will be 3xx to make the sorting of this data easier for the counties.

Reports Overview

This section describes the requirements for the reports generated from the TRAC Application. Data for each report will be delivered in an electronic file format. Counties will have the ability to use the files to create their own reports. These report files will be generated monthly and will report individuals who are within a predetermined time period of the specified time limit.

Approaching Clocks Report (Existing)

The system shall provide a report file that produces a summary and detailed list of active individuals, by CIN or County ID, who are in month 54 or month 58 of the 60 month time limit for the State (CalWORKs) and/or federal (TANF) Programs; and/or in month 15/21 of the 18/24 month time limit for Welfare to Work. Counties may continue to receive this existing Approaching Clocks Report file until they begin receipt of the new 48/53-60 report described below.

48/53-60 Approaching Clocks Report (New)

This system shall provide a report file that produces a summary (provided to CDSS) and detailed list (provided to the counties) of individuals who are in one of the following months of the 60-month time limit for the State (CalWORKs) and/or federal (TANF) Programs: 48, 53, 54, 55, 56, 57, 58, 59, or 60; and/or in month 15/21 of the 18/24 month time limit for Welfare to Work.

These reports will be used by caseworkers as a caseload management tool to ensure that upcoming time on aid limits are recognized and appropriately controlled.

Exceeding Clocks Report

The system shall provide a report that produces a summary and detailed list of active individuals who have exceeded the 60-month time limit on the TANF 60-month, CalWORKs 60-month and Welfare to Work 18/24-month time on aid limits. These reports will be generated monthly and will report individuals who have exceeded specified time limits.

- ☐ The TANF 60-month report file will include all active individuals whose TANF months used clock is equal to or greater than 60
- ☐ The CalWORKs 60-month report file will include all active individuals whose CalWORKs months used clock is equal to or greater than 60
- ☐ The WTW 18/24-month report file will include all active individuals whose Welfare to Work months used clock is equal to or greater than 18 or 24

This report can be used by caseworkers to ensure that individuals who have exceeded time on aid limits meet one or more time extender criteria or are the case handled as per applicable regulations.

Managers can also use this report to help ensure that individuals who are no longer eligible to assistance due to expiration of time limits are appropriately notified and appropriate action taken.

Monthly Projections Report

The system shall produce a report based on a monthly projection of the number of active individuals who will exceed the TANF 60-month, CalWORKs 60-month and Welfare to Work 18/24-month time on aid limits within the next 6 months. These reports will be generated monthly and report individuals who will exceed specified time limits by month for the upcoming 6 months.

- ❑ The TANF 60-month report file will include the number of active individuals who have used 54, 55, 56, 57, 58 or 59 months on their TANF 60-month time clock and give summary totals for each month
- ❑ The CalWORKs 60-month report file will include the number of active individuals who have used 54, 55, 56, 57, 58 or 59 months on their CalWORKs 60-month time clock and give summary totals for each month
- ❑ The WTW 18/24-month report file will include the number of active individuals who have used 12, 13, 14, 15, 16 or 17 months of their 18-month time limit; and individuals who have used 18, 19, 20, 21, 22 or 23 months of their 24-month time limit and give summary totals for each month

County managers can use this report to project the number of individuals who may be discontinued from aid or potentially transitioning to community service activities.

Multiple County Aid Report

The system shall produce a report detailing the number of individuals who have received aid in multiple counties over time. This is a statewide aggregate report that will be generated monthly

Refer to the specific report file layouts in the **Attachment** sections of this document for report layouts.

6. Development Process

The cornerstone of the TRAC Application is data. Without accurate, complete, and timely data from the counties, the TRAC Application will fail to be a useful tool for eligibility workers across California. Given the criticality, the process of loading data from the counties into the TRAC Application is one of the most important steps in the development process. This process includes the following steps:

- ☐ Review of source system data
- ☐ Creation of extraction/loading programs
- ☐ Testing of extraction data
- ☐ Trial load
- ☐ Initial conversion
- ☐ Ongoing data extraction

Each step and the associated timeline will be described over the following pages.

Review of Source System Data

The review of data directly from the eight systems that will provide the primary data for the TRAC Application is one of the first components of the development process. This step is designed to provide the WDTIP Team with an understanding of how data currently resides in the source systems. This process may identify some data integrity issues early on and/or confirm the programming approach assumptions that are critical to the development of both the extraction and load programs. For example, a county may be using the program end date field in their system to indicate a client's death by typing in the words "Death Date" in the date field. This additional text could be problematic for the load program to understand and cause error in testing later in the project.

The following are the necessary tasks, responsible parties, and due dates associated with this step:

<i>Task</i>	<i>Responsible Party</i>
Provide sample data in ASCII text format from each source system	<input type="checkbox"/> ISAWS <input type="checkbox"/> WCDS <input type="checkbox"/> LEADER <input type="checkbox"/> Ventura <input type="checkbox"/> Stanislaus <input type="checkbox"/> Riverside <input type="checkbox"/> Merced <input type="checkbox"/> San Bernardino
Review data and make appropriate programming modifications	<input type="checkbox"/> The WDTIP Team

Creation of Extraction/ Loading Programs

The creation of the extraction and loading programs to transfer the data from the county systems to the TRAC database is the largest task in the development process. To meet this goal, each county/consortium will need to map data from its systems to a standard file format (see Standard File Format). This activity will include not only providing the data in a standard process (i.e., two transactions to provide a start and end date for a program) but also will need to convert any values in the county system into the values described in the Reference Codes section of this deliverable. Finally, the county/consortia will need to insure that only changes to the data elements requested by the WDTIP Team are sent in the daily files (as opposed to all records in the database). The level of effort involved in this activity should not be underestimated.

To assist the counties/consortia in this process, the WDTIP will provide a number of opportunities for testing of these files (and the program designed by the WDTIP to load the data). These testing files should contain a manageable number of records from each county to insure the data can be properly loaded and that after it is loaded, it is represented in the correct way to the counties. As the files are loaded, exception files will be sent to the counties describing any problems incurred. Additionally, the WDTIP Team will work with the extraction file creators to identify potential ways to improve the programming for this transfer. This program modification may focus on the creation of the extraction file or on the loading of the data. This is an iterative process. It may be necessary to complete this testing process multiple times for a county/consortia. The testing will continue until both the creator of the extraction file and the WDTIP Team are both satisfied that the data will be successfully loaded into the TRAC database.

The following are the necessary tasks, responsible parties, and due dates associated with this step:

<i>Task</i>	<i>Responsible Party</i>
Design, develop and test (internally) extraction programs	<input type="checkbox"/> ISAWS <input type="checkbox"/> WCDS <input type="checkbox"/> LEADER <input type="checkbox"/> Ventura <input type="checkbox"/> Stanislaus <input type="checkbox"/> Riverside <input type="checkbox"/> Merced <input type="checkbox"/> San Bernardino
Develop and test data loading programs	<input type="checkbox"/> The WDTIP Team
Send test extraction files (iterative until agree successful)	<input type="checkbox"/> All counties
Send exception reports and programming modification suggestions (iterative)	<input type="checkbox"/> The WDTIP Team

Trial Loads

Once the testing is completed and both the counties and WDTIP are satisfied that the programs on both ends of the data transfer are working properly, a trial load will be completed for each county's data. This trial load will be a complete data transfer from the source system. The purpose of this trial load is to give the counties a clear understanding of the effort necessary to modify the data loading into the TRAC database. It will also give the creator of the extraction file an understanding of the processing time required to extract the data. Once the WDTIP has received this file, an exception file will be sent back to the originator of the data. This file can then be used as a guide to fix any remaining data/programming problems. For example, using the scenario from the ***Review of Source System Data***, the "Death Date" text may be necessary to remove from the records in the database. This may require some manual fixes or programs to correct errant data.

The following are the necessary tasks, responsible parties, and due dates associated with this step:

<i>Task</i>	<i>Responsible Party</i>
Generate trial load (in standard file format with initial load records)	<input type="checkbox"/> All counties
Send exception reports and programming modification suggestions	<input type="checkbox"/> The WDTIP Team

Initial Conversion

Once all of the programming and data corrections have been made, the initial conversion of each county's data will begin. This step includes both the creation and transmission of the first extraction file (containing all of the historical data for an individual from each county) and the beginning of the daily updates that will be necessary from the source system. Again, after the file has been received, any records that cannot be loaded in the database will be noted in the exception report. From this point forward, it will be the responsibility of the county to correct data that is listed in the exception report.

The following are the necessary tasks, responsible parties, and due dates associated with this step:

<i>Task</i>	<i>Responsible Party</i>
Generate initial conversion load	<input type="checkbox"/> All counties
Send exception reports	<input type="checkbox"/> The WDTIP Team
Correct data and send back to the WDTIP	<input type="checkbox"/> All counties

Ongoing Loads (After Initial Conversion)

After the data for a county has been converted, a regular, daily update must be sent to update the TRAC Application with new or modified data. These ongoing loads should only contain changes or additions to data that currently exists in the TRAC Application. For example, if an individual has an opened CalWORKs program, and the start date is contained in the initial load, it is not necessary to send that information again. Only when the individual's program is changed or closed should that program participation information be sent.

The following are the necessary tasks, responsible parties, and due dates associated with this step:

<i>Task</i>	<i>Responsible Party</i>
Generate ongoing conversion load	<input type="checkbox"/> All counties
Send exception reports	<input type="checkbox"/> The WDTIP Team
Correct data and send back to the WDTIP	<input type="checkbox"/> All counties

7. WDTIP Technical Contacts

This section contains contact information for members of the WDTIP Team who can be contacted to obtain further clarification on any of the topics that are covered in the External Developer's Guide. The contact information as well as the areas of expertise for the WDTIP Team members is contained in the following table.

<i>Area of Expertise</i>	<i>Contact Name</i>	<i>Phone Number</i>	<i>E-mail Address</i>
Screens and Reports/ Conversion	Loy Holder	(916) 229-4477	lholder@sid.hhsdc.ca.gov
	Grace Kajita	(916) 229-4412	gkajita@sid.hhsdc.ca.gov
	Kim Murdock	(916) 229-4485	kmurdock@sid.hhsdc.ca.gov
System Technical Support (Database, Protocols, etc)	Brian Kick	(916) 229-3380	bkick@sid.hhsdc.ca.gov

For other questions regarding the WDTIP, please contact the WDTIP Help Desk at (877) 365-7378.

Note: the fax number for the WDTIP Team members is (916) 229-4487.

8. Issues and Assumptions

This section documents all of the issues and assumptions related to design. These assumptions were recorded during the Joint Application Design (JAD) sessions and during subsequent meetings.

Issues

Some questions regarding the design of the TRAC Application were not able to be completely resolved during the Design Phase of the WDTIP. To insure that these issues are not left unanswered during development, they have been listed in the following table.

<i>Issue #</i>	<i>Description</i>
1	Confidential cases, such as a relative of an employee who is receiving aid in a county, may need to be limited in access to statewide users. The impact of this issue is being researched with the user community.
2	The exact format and use of the Account Code data element has not been determined. This value was originally designed for charge back purposes. Further definition of this field will need to be provided to the county/consortia developers if it is a required value from the county systems.
3	An assumption (Assumption #25 below) has been made that case level aid codes will not be received directly from the counties. However, CDSS is researching the impact of case level aid codes on individual time clocks.
4	The proposed loading logic for exemptions will not allow for concurrent exemption periods. Counties may need the ability to have concurrent exemption periods. Analysis is being completed to determine if it is necessary, and how that functionality will impact the application design.

Assumptions

The assumptions will be used to guide the development process.

Assumption #	Description	Source
1	The data that originated in the county systems and loaded into MEDS has not been modified from its original form. MEDS sorts the data then sends it to systems with which it interfaces, such as SIS.	Design Process
2	An automated tool for conversion (e.g., ACL) will not be used since the mapping to the standard extraction file format will be completed by each county/system.	Design Process
3	<p>The following is a description of the MEDS processing procedures as they are understood by the WDTIP:</p> <ol style="list-style-type: none"> 1. County/Consortia sends MEDS extract file. 2. MEDS rejects a whole record if any of the data contain in the MEDS extraction file does not meet MEDS exception criteria. 3. MEDS sends a hard copy report to the counties containing each reject record. 4. Counties are responsible for updating MEDS directly so that both systems are synchronized. 5. At the end of a period (month or quarter), the county/consortia sends a snapshot of all the data contained in their systems which is required by MEDS. 6. MEDS matches the data from the counties/consortia and produces a reconciliation report for unsynchronized data between MEDS and the county system. 7. Counties are responsible for manually updating data on MEDS that appears in the reconciliation report. 	Design Process
4	Counties have a very low number of diversion program participation cases. Due to the low number of cases, diversion information is recorded through manual procedures that are different from county to county.	JAD Session

Assumption #	Description	Source
5	LA Legacy did not capture any data associated with diversion payments. LEADER does not currently (7/28/99) have any data loaded for diversion payments. The diversion information shall be captured in the future and can be provided to TRAC.	JAD Session
6	In Los Angeles, Welfare to Work Start Date is currently stored in GEARS. In the future, LEADER has a placeholder in their database for storing that data.	JAD Session
7	Non-California data will not be converted from LA Legacy to LEADER (it did not exist in LA Legacy). However, LEADER has the capability to store this information.	JAD Session
8	The JAD participants agreed that data not available from the county systems (non-converted data) shall be populated from data currently residing in SIS.	JAD Session
9	Data that is currently populated in SIS does not contain complete information to calculate the time clocks. Specifically, exception, diversion, supportive service, child support reimbursement and some individual information is not stored in SIS.	Design Process
10	All source systems need to provide either the CIN or all of the following fields: County Code, County Serial Number, FBU, Person Number, Last Name, Middle Name, First Name, SSN, Date of Birth and Gender. If TRAC does not receive this data, the entire individual's record shall not be loaded into the database.	Design Process/ Meeting with DHS
11	Application programs that are modified from the current Pre-SAWS application shall continue to use the same Program IDs and therefore, not meet the program naming standards outlined in the Design and Coding Standards document.	Design Process

Assumption #	Description	Source
12	All jobs shall be named using the Development Environment as the target environment as it will be the first environment in which the jobs will be run.	Design Process
13	ABAWD data shall continue to be provided in the MEDS application. Since this information will exist in MEDS, the ABAWD data will not be stored in the TRAC database.	WDTIP Project Management verified by JAD participants
14	If the counties do not send data for a period of days (due to the MEDS reconciliation process or other reasons), they shall either provide the daily updates that were missed in a single file or in a series of files numbered sequentially.	Design Process
15	The counties shall continue to provide program participation information for all program types including CalWORKs, Food Stamps, Medi-Cal, Foster Care, etc.	WDTIP Project Management Decision
16	SCI will use a matching algorithm to link an individual to a CIN through demographic information. This algorithm is initiated by the program B3SC010 and returns the results through B3SC020.	Design Process
17	In the JAD sessions, the conversion workgroup participants modified an earlier assumption that Welfare to Work Start Date could not be re-initiated. The new assumption is that the counties would provide the Workplan Sign Date, and the most recent information received would be the date used to start the Welfare to Work 18/24-Month Time Clock.	JAD Session (this assumption re-defines an older assumption from the Updated Business Requirements Document V04 [Assumption # 27 in this document])
18	In the JAD sessions, the application workgroup verified that a participant is active in a program when no program end date is present.	JAD Session (Application Workgroup)
19	Report information shall be distributed via electronic file.	JAD Session (Application Workgroup)

Assumption #	Description	Source
20	In the JAD sessions, the application workgroup agreed that each county's data processing center would link demographic and other data to report file information. Therefore, there is no need to send demographic information in the report file.	JAD Session (Application Workgroup)
21	The Multiple County Aid Report shall not be generated until all county systems are converted. This decision was later reversed, and currently WDTIP is generating this report for all converted counties.	JAD Session (Application Workgroup)
23	Counties shall not send minor consent program participation records to the TRAC application.	Design Process
24	The TRAC application will only keep a single instance of a supportive service only payment for an individual in a county. If multiple records are sent with a status of "AID", the system will only display one record. Additionally, to remove that record, a county should only send one record for that supportive service only payment month with a status code of "DISC".	JAD Sessions (Technical Workgroup)
25	Only individual aid codes should be sent to TRAC. If the county source system uses case level aid codes (i.e., "3E" after 10/1/1999), the county's extraction file program should translate that value into the appropriate individual aid code.	Design Process (from ACL 99-54)
26	The age eligibility parameters for CalWORKs will also apply to TANF with the exception of minor parents. Minor parents are excluded from the CalWORKs time clock requirements but time is counted toward the TANF time clock for minors who are heads-of-households, or married to heads of households.	Updated Business Requirements Document V04

Assumption #	Description	Source
27	The 18/24-month clock does not start until the Welfare-to-Work plan is signed (i.e., the Welfare-to-Work plan initial sign date or the Welfare to Work plan refuse to sign date).	Updated Business Requirements Document V04
28	Each time clock has its own set of business rules.	Updated Business Requirements Document V04
29	Receipt of supportive services only (no aid payment) counts as a month toward the TANF 60-month clock unless the recipient is employed. Receipt of supportive services only (no aid payment) does not count as a month toward the CalWORKs 60-month or 18/24-month clocks.	Updated Business Requirements Document V04
30	The TANF start date is based upon the date of issuance of TANF funds in California.	Updated Business Requirements Document V04
31	With the exception of litigation, there is never an occasion where the 18/24-month clock is reinitiated.	Updated Business Requirements Document V04
32	The JRP workgroup expressed an interest in capturing and displaying the reason code for diversion payment. The item was determined to be out of project scope and was subsequently left out of the requirement list. However, the JRP workgroup would like to have it be considered as a future enhancement to the TRAC Application.	Updated Business Requirements Document V04
33	Time clock tracking calculations for diversion payments are individually based. When a diversion payment is made, the appropriate clock will tick for each aided adult in the case.	Updated Business Requirements Document V04
34	WDTIP will not track Homeless Assistance payments for the Once-in-a-Lifetime Homeless limitation, but will instead track the effect of the receipt of homeless benefits on the time clocks.	Updated Business Requirements Document V04
35	Security of the TRAC Application will be applied at the application level.	Updated Business Requirements Document V04

Assumption #	Description	Source
36	The TRAC Application will track ABAWD information even though the information is already present in the MEDS system. In addition, the JRP workgroup expressed the desire to minimize the impact of the creation of the extraction files for ABAWD. See Project Tracking System Issue# 17 for revision of assumption.	Updated Business Requirements Document V04; Assumption #13
37	The WDTIP Team will research to determine whether ABAWD information should be converted from MEDS to the TRAC database and incorporate the solution into the design of the TRAC Application. See Project Tracking System Issue# 17 for revision of assumption.	Updated Business Requirements Document V04; Assumption #13
38	Data which can be modified through the county system's batch processes will not be updateable online in the TRAC Application. Details on which fields may be updated and the associated security issues will be determined during the Joint Application Design (JAD) sessions.	Updated Business Requirements Document V04
39	The JRP workgroup expressed an interest in having, at a minimum, an indicator showing Statewide overissuance/overpayment information at an individual level and the county in which it was initiated. The item was determined to be out of project scope and was subsequently left out of the requirements' list. However, the JRP workgroup would like to have it be considered as a future enhancement to the TRAC Application.	Updated Business Requirements Document V04

Assumption #	Description	Source
40	The JRP workgroup expressed an interest in having the TRAC Application capture and display case participation data. It was determined that the TRAC Application will capture the county's case number and display it at an individual level. Capturing and displaying other additional fields as it relates to case participation at the case level was determined to be out project scope. However the JRP workgroup would like to have it be considered as a future enhancement to the TRAC Application.	Updated Business Requirements Document V04
41	The JRP workgroup concluded that it was not feasible to collect data on fleeing and drug felon applicants. The item was subsequently left out of the requirement list.	Updated Business Requirements Document V04
42	The WDTIP Team will work with Project Management, DHS and other welfare consortiums to determine whether the topic of data retrieval for LEADER and ISAWS applications during Statewide file clearance (known-to-welfare search) is within the scope of WDTIP.	Updated Business Requirements Document V04
43	The JRP workgroup concluded that Job Retention Services was a local issue and agreed to leave it out of the requirements' list.	Updated Business Requirements Document V04
44	WDTIP will evaluate any new state regulations up until the approval of the Updated Business Requirements Document for business requirements and until August 31, 1999 for technical requirements. Any new legislation after the cut-off time will be managed according to the Scope and Change Control Process in the Project Management Deliverable.	Updated Business Requirements Document V04
45	WCDS and the C-IV counties do not use the CIN as a unique client identifier in their legacy systems.	Updated Business Requirements Document V04

Assumption #	Description	Source
46	The project will be developed, tested, and implemented on an OS/390 compatible mainframe computer.	System Architecture Model V02
47	The computing systems of the Health and Human Services Data Center (HHSDC) will be used to support the development, testing, and implementation of the application.	System Architecture Model V02
48	The central database of the application will be implemented on IBM's DB2 database platform.	System Architecture Model V02
49	All database storage will be contained within the DB2 capacity of the HHSDC. No distributed processing incorporating the county databases will be used.	System Architecture Model V02
50	End users will gain access to the application database through CICS transactions running on 3270 or 3270 emulation terminals.	System Architecture Model V02
51	Deloitte Consulting shall have full staff leadership and management responsibilities for the WDTIP and will submit their recommendations to the State WDTIP management staff for approval.	WDTIP Statement of Work, Section 5
52	Deloitte Consulting will report directly to the SAWS Deputy Director or his representative.	WDTIP Statement of Work, Section 5
53	The State will provide on a timely basis, as defined by the project workplan, qualified State resources to assist in the design, development and implementation of the Welfare Data Tracking Implementation Project system. These resources are identified in the project organization charts in the IAPDU, Section 4.3.	WDTIP Statement of Work, Section 5
54	The State will provide timely resolution, as defined by the project workplan, of State controlled issues that affect the project plan and schedule.	WDTIP Statement of Work, Section 5

Assumption #	Description	Source
55	The counties will provide adequate resources to participate in Requirements Validation, Data Conversion, Training, and Implementation activities.	WDTIP Statement of Work, Section 5
56	Individual counties will be responsible for conversion data mapping and extraction activities. In addition, counties will provide necessary test data to the project on a timely basis. These activities will be completed on time and in accordance with the county's rollout date.	WDTIP Statement of Work, Section 5
57	All data files provided by the counties for conversion and on-going loads will be Year 2000 compliant.	WDTIP Statement of Work, Section 5
58	User training will be conducted utilizing the train-the-trainers approach. Some training may be regionally based. Training will be conducted at existing county or State facilities. Deloitte Consulting will not be responsible for providing any training equipment.	WDTIP Statement of Work, Section 5
59	The majority of system-generated reports will be handled under a separate project. Hence, there will be approximately 10 internal audit reports that the Welfare Data Tracking Implementation Project must generate.	WDTIP Statement of Work, Section 5
60	An Advisory Committee will be formed including representatives from the California Department of Social Services (CDSS), the Department of Information Technology (DOIT), the Health and Human Services Data Center (HHSDC), the County Welfare Directors Association (CWDA), the Department of Health Services (DHS), the Welfare Data Tracking Implementation Project IV&V Vendor, and Deloitte Consulting.	WDTIP Statement of Work, Section 5

Assumption #	Description	Source
61	Although the State is planning for the Statewide Client Index (SCI) management and maintenance to be transferred from DHS to WDTIP sometime during the duration of the project, the SCI system and it's maintenance will be the sole responsibility of the State.	WDTIP Statement of Work, Section 5
62	The WDTIP Phase II schedule and deliverable due dates will be reviewed and revised, as appropriate, as a component of the Phase II Workplan deliverable. The WDTIP Phase III schedule and deliverable due dates will be reviewed and revised, as appropriate, as a component of the Phase III Workplan deliverable.	WDTIP Statement of Work, Section 5
63	The WDTIP user training delivered to all of the California counties will be based upon the training of a selected county trainer. User training will be delivered by the Contractor to the county trainers at regional sites using existing county training facilities at no cost to either the Contractor or the State. Training will be delivered through classroom instruction using appropriate training materials to be provided by the Contractor and further defined in the implementation strategy deliverable. Technical support will be provided by the Contractor to county technical resources. The manner and duration of the training will be determined during the development of the Training Strategy.	WDTIP Statement of Work, Section 6

Assumption #	Description	Source
64	<p>Counties will provide resources in a timely manner, as defined by the project workplan, to assist in the following activities:</p> <p>Requirements Validation (June/July/On-Going) Responsible for:</p> <ul style="list-style-type: none"> • Initial Requirements Validation • On-Going Requirements Validation <p>Implementation Coordinator (June/July/On-Going) Responsible for:</p> <ul style="list-style-type: none"> • Implementation Schedule • Primary County Contact • Determine other County Contacts • Communication within the County • Change Leadership for County Business Processes • User Acceptance Testing <p>Data Conversion (June/July/On-Going/Technical Person) Responsible for:</p> <ul style="list-style-type: none"> • Defining what is Automated • Defining what is Manual • Data Mapping • County Batch Program Development • Testing and Trial Conversion • County Acceptance Testing • Initial Batch Load • On-Going Load <p>Training (Time requirements adjacent to tasks) Responsible for :</p> <ul style="list-style-type: none"> • Design of County Training Approach (July 99) • Review of Training Materials (Oct 99) • Scheduling (March 00/On-Going) • Receiving Training (March 00/On-Going) • Trainer for County (March 00/On-Going) 	WDTIP Statement of Work, Section 6

Assumption #	Description	Source
65	The WDTIP Advisory Committee will meet on a regular basis to assist in reviewing project progress and aid in maintaining project scope and progress.	WDTIP Statement of Work, Section 6
66	Since all funds were included in the budget, the control agency review scheduled to occur upon completion of each of the three project phases, will result in authorization to begin the next phase of the project. The control agency review process will start early and not result in any lapse between the identified project phases.	WDTIP Statement of Work, Section 6
67	The Contractor will be fully responsible for maintaining appropriate staffing levels, providing staff leadership and discharging staff and project management responsibilities.	WDTIP Statement of Work, Section 6
68	Designated key personnel will be on site throughout the design, development, and implementation phases of the WDTIP project, in accordance with Section 9.1 of this Agreement.	WDTIP Statement of Work, Section 6
69	The Contractor has the discretion to hire consulting staff of their choice; however, they will attempt to use those consultants with previous SAWS-TA experience. Additionally, resumes must be submitted and approved by State management before any consultant commences work on the WDTIP.	WDTIP Statement of Work, Section 6
70	All required batch jobs will fit within prescribed batch windows to be mutually determined during the appropriate project phase.	WDTIP Statement of Work, Section 6
71	The database shall be converted to the physical data model structure described in this document prior to the implementation of any of the batch programs.	Design Process
72	Data shall be received on a daily basis from the counties, MEDS, and SCI.	JAD Session (Technical Workgroup)

Assumption #	Description	Source
73	The following MEDS programs shall not be used after the database has been reconfigured to the physical data model structure described in this document: B3DU004, B3DU010, B3DU011, B3DU012, B3DU013, B3DU014, and B3DU017.	Design Process
74	No Batch Headers shall be provided for files interacting with SCI or MEDS (File names SCI-INPUT-FILE, SCI-OUTPUT-FILE, SCI-COMB-FILE, MEDS-EXT-FILE).	Design Process/ Meeting with DHS
75	The daily job flow shall be completed successfully prior to beginning the monthly job flow.	Design Process
76	Job B3DDL01 and B3DDL02 can be run concurrently, but B3DDL03 must not be run until B3DDL01 and B3DDL02 have been completed successfully.	Design Process
77	B3DDU01 must not be run until all other daily jobs have been completed.	Design Process
78	In the CIN-COMB-FILE provided by SCI, a Secondary CIN cannot be linked to more than one primary CIN. If this happens, the record shall not be processed in the database.	Design Process
79	All counties will send only individuals to TRAC in the extraction file. All payees, organizations, facilities, etc. shall need to be cleansed by the counties prior to sending the file.	Design Process/ Meeting with DHS (August 12, 1999)
80	Reports shall be generated in an e-file format.	JAD Session (Application Workgroup)
81	SCI will use a consistent matching algorithm so that regardless of the source of data sent for CIN generation (either MEDS or SIS), only one CIN shall be generated for an individual.	Design Process/ Meeting with DHS (August 12, 1999)
82	Counties are responsible for the design, development, and testing of their extraction files.	JAD Session (Technical Workgroup)

Assumption #	Description	Source
83	ABAWD was initially identified as a business requirement, but after further discussion and approval from CWDA it was decided to not change its existing functionality. See Project Tracking System Issue# 17.	Updated Business Requirements Document V04; Assumption #13

Attachment A – Extraction File Layout

Introduction

The Extraction file is composed of different columns that qualify each of the fields in the extraction file. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ ***Mandatory/Optional:*** This column categorizes each of the fields in the extraction file into a mandatory or optional field. In some cases some fields are categorized as mandatory/optional. For further clarification for such fields please refer to the Rule column
- ❑ ***Data Length:*** This column will display the length of the field in the extraction file
- ❑ ***Data Type:*** This column will display the data type each of the fields in the extraction file such as for example: A/N (Alpha Numeric), N (Numeric), and Date
- ❑ ***Definition:*** This column will display the definition of each field. If the field is a TRAC database stored field, the definition provided will be the same as the one provided in the Data Dictionary section of the Developer Notes
- ❑ ***Rule:*** This column will display additional information that is necessary to qualify each file in the extraction file. For each field one or more of the following components may be included under this column
 - **Rule:** provides any additional validation rules that are special to a field. For example: The Program Discontinuance Reason Code is optional unless the Program Status Code is equal to 'Disc'
 - **Sample Data:** provides a sample value of the data that could be contained in the field. For example: the field Batch Number could contain '0001' as a valid value
 - **Valid Value:** provides the values that will need to be included for each field. If the field contains values that are found in a reference table, the name of the table and the column where the field is located will be provided. For example: the field County Code will be found under, WDTIP Reference codes (Table: County Code, Column: County Code). This field will also contain the valid format for the specific field such as the case for date and time type of fields

In addition, the following rules should be observed with respect to file transmission:

- All records are of fixed length 150 bytes. If the record length is less than 150 bytes a filler should be inserted calculated as 150 bytes minus record length.
- The WDTIP system expects one extract file per county per day.
- The batch version number should start with 0 (zero).

The COBOL file layout is provided for reference.

Attachment B – Exception File Layout

Introduction

This exception file is composed of different columns that qualify each of the fields. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ *Data Length:* This column will display the length of the field
- ❑ *Data Type:* This column will display the data type each of the fields such as for example: A/N (Alpha Numeric), N (Numeric), and Date
- ❑ *Definition:* This column will display the definition of each field
- ❑ *Rule:* This column will display additional information that is necessary to qualify each file in the report file. For each field one or more of the following components may be included under this column
 - Rule: provides any additional validation rules that are special to a field
 - Sample Data: provides a sample value of the data that could be contained in the field
 - Valid Value: provides the values that will need to be included for each field

Attachment C – Report Layout Approaching Clocks

Introduction

This report file is composed of different columns that qualify each of the fields. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ *Data Length: This column will display the length of the field*
- ❑ *Data Type: This column will display the data type each of the fields such as for example: A/N (Alpha Numeric), N (Numeric), and Date*
- ❑ *Definition: This column will display the definition of each field*
- ❑ *Rule: This column will display additional information that is necessary to qualify each file in the report file. For each field one or more of the following components may be included under this column*
 - Rule: provides any additional validation rules that are special to a field
 - Sample Data: provides a sample value of the data that could be contained in the field
 - Valid Value: provides the values that will need to be included for each field

Attachment C.1 – Report Layout 48/53-60 Approaching Clocks

Introduction

This report file is composed of different columns that qualify each of the fields. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ *Data Length: This column will display the length of the field*
- ❑ *Data Type: This column will display the data type each of the fields such as for example: A/N (Alpha Numeric), N (Numeric), and Date*
- ❑ *Definition: This column will display the definition of each field*
- ❑ *Rule: This column will display additional information that is necessary to qualify each file in the report file. For each field one or more of the following components may be included under this column*
 - Rule: provides any additional validation rules that are special to a field
 - Sample Data: provides a sample value of the data that could be contained in the field
 - Valid Value: provides the values that will need to be included for each field

Attachment D – Report Layout Exceeding Clocks

Introduction

This report file is composed of different columns that qualify each of the fields. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ *Data Length:* This column will display the length of the field
- ❑ *Data Type:* This column will display the data type each of the fields such as for example: A/N (Alpha Numeric), N (Numeric), and Date
- ❑ *Definition:* This column will display the definition of each field
- ❑ *Rule:* This column will display additional information that is necessary to qualify each file in the report file. For each field one or more of the following components may be included under this column
 - Rule: provides any additional validation rules that are special to a field.
 - Sample Data: provides a sample value of the data that could be contained in the field
 - Valid Value: provides the values that will need to be included for each field

Attachment E – Report Layout Multiple County Aid

Introduction

This report file is composed of different columns that qualify each of the fields. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ *Data Length:* This column will display the length of the field
- ❑ *Data Type:* This column will display the data type each of the fields such as for example: A/N (Alpha Numeric), N (Numeric), and Date
- ❑ *Definition:* This column will display the definition of each field
- ❑ *Rule:* This column will display additional information that is necessary to qualify each file in the report file. For each field one or more of the following components may be included under this column
 - Rule: provides any additional validation rules that are special to a field
 - Sample Data: provides a sample value of the data that could be contained in the field
 - Valid Value: provides the values that will need to be included for each field

Attachment F – Report Layout Monthly Projections

Introduction

This report file is composed of different columns that qualify each of the fields. In order to clarify the information that will be contained in each column, an overview of each column is provided.

- ❑ *Data Length:* This column will display the length of the field
- ❑ *Data Type:* This column will display the data type each of the fields such as for example: A/N (Alpha Numeric), N (Numeric), and Date
- ❑ *Definition:* This column will display the definition of each field
- ❑ *Rule:* This column will display additional information that is necessary to qualify each file in the report file. For each field one or more of the following components may be included under this column
 - Rule: provides any additional validation rules that are special to a field
 - Sample Data: provides a sample value of the data that could be contained in the field
 - Valid Value: provides the values that will need to be included for each field

Attachment G – TRAC Data Model

Introduction

This attachment includes the physical data model of the TRAC Application. It provides the foundation for the table structure that will be implemented in the TRAC database. The physical data model is a graphical representation of all the tables in the TRAC Application with their corresponding columns and relationships.

Attachment H – TRAC Data Dictionary

Introduction

The data dictionary contains a complete listing of the tables and columns that make up the TRAC database. The following components are included in the data dictionary:

- ❑ Table Name
- ❑ Table Description
- ❑ Column Name
- ❑ Column Description
- ❑ Column Datatype
- ❑ Column Null Option (value = “Null” or “Not Null”)
- ❑ Column Default Value (Note: In instances where the column default is system generated such as Current Date/Time, the value description in the data dictionary will be noted as “WITH DEFAULT”. For character fields, the “WITH DEFAULT” designation will mean that the field can have a null value given that all DB2 fields, by convention are set to be “Not Null”).

Attachment I – TRAC Reference Codes

Introduction

This section provides a description of the codes that will reside in the TRAC database application. The purpose for identifying these codes is to define a statewide standard for the codes that county systems will use to send data to the TRAC Application.

For each reference code, the information presented is divided into the following subsections:

- ☐ Business Description
- ☐ Business Need
- ☐ Information Sources
- ☐ Standard Values
- ☐ Comment